The Simulation of the Dynamic Link Allocation Router (DyLAR)

Wei Song

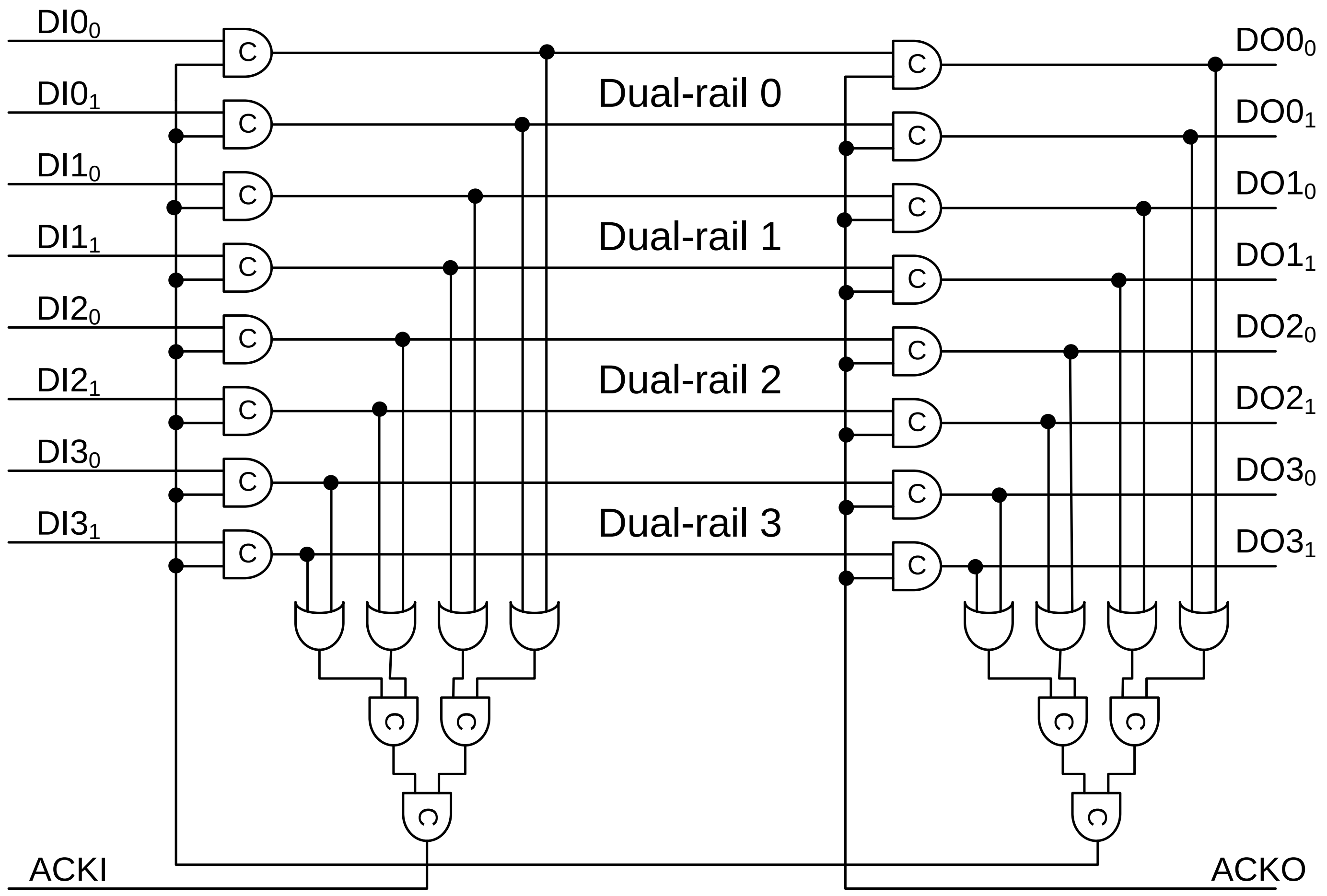
Advanced Processor Technology Group
The School of Computer Science

2014/5/13

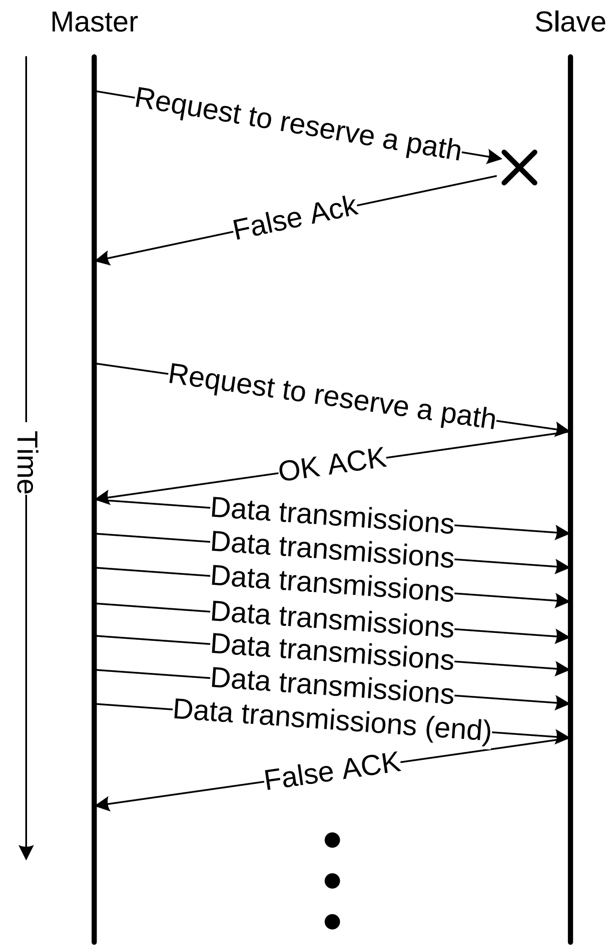
Overview

- A brief review of the *Dynamic Link Allocation* flow control method
- The new simulation platform
- Some simple performance analyses
- An alternative method of the *task request procedure*
- Future schedule

Serial is better than Parallel

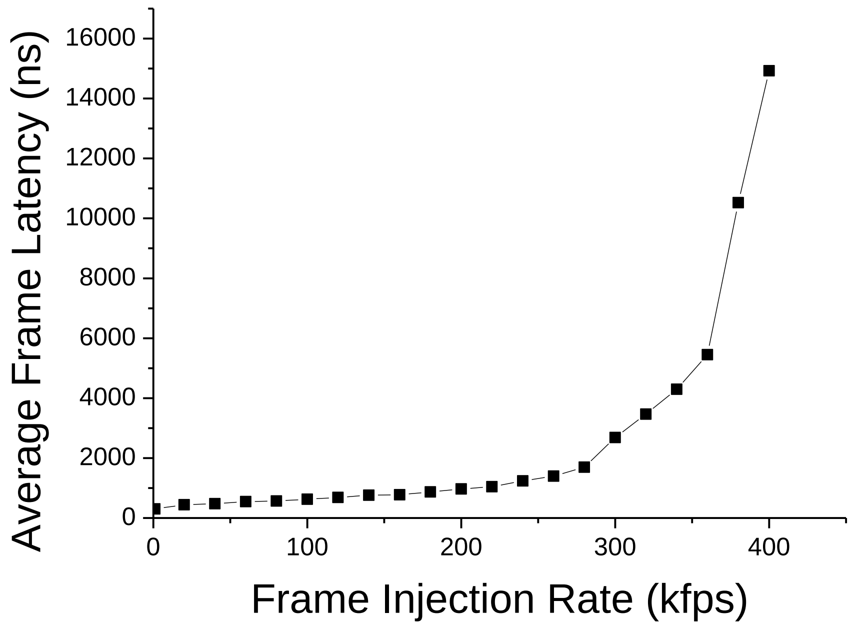


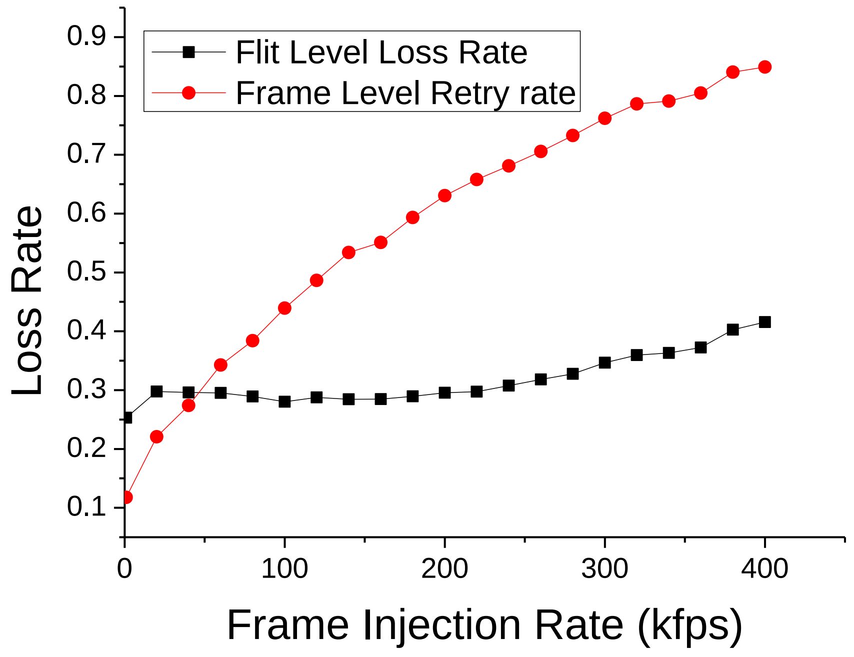
Bandwidth efficiency is less than 50%



The high Loss Rate

Simulation results of a 6x6 NoC.





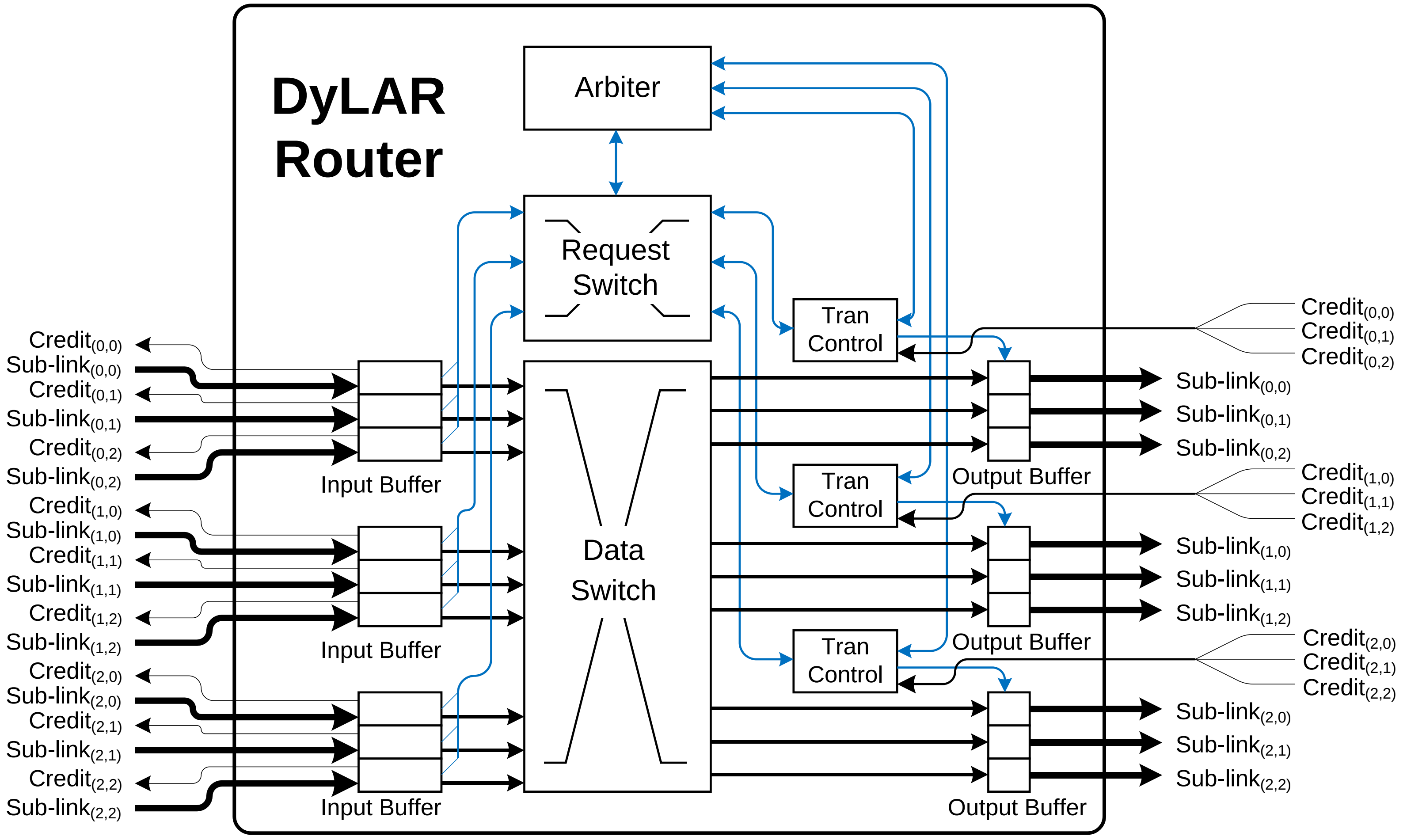
Advanced Processor Technology Group
The School of Computer Science

2014/5/13

Some hypotheses of *DyLAR*

- Asynchronous circuits prefer serial rather than parallel channels
- Connection oriented communications only have a bandwidth efficiency less than 50%
- The high retry rate of connection oriented communication is reducible by add virtual channels
- The input buffer could be smaller than flit size when using serial channels

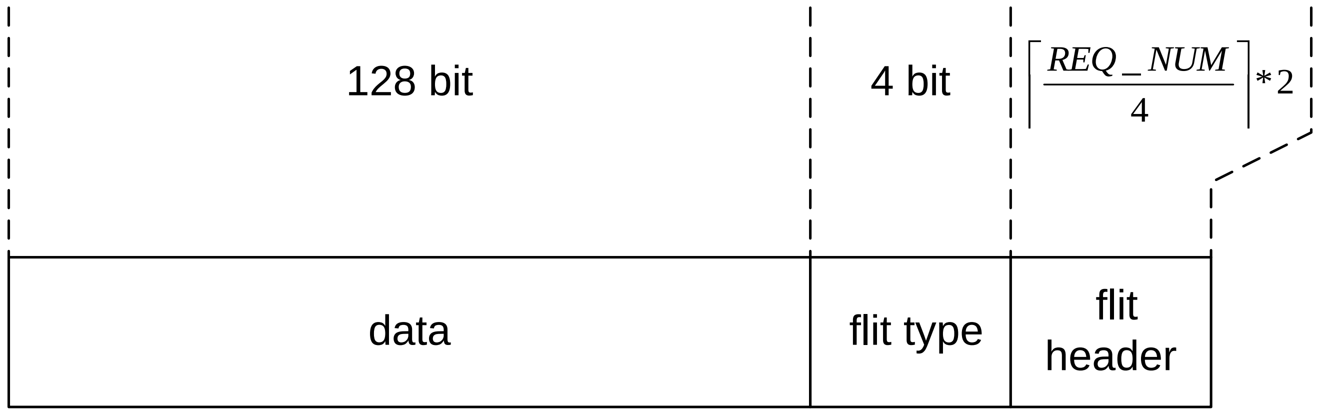
The DyLAR Router

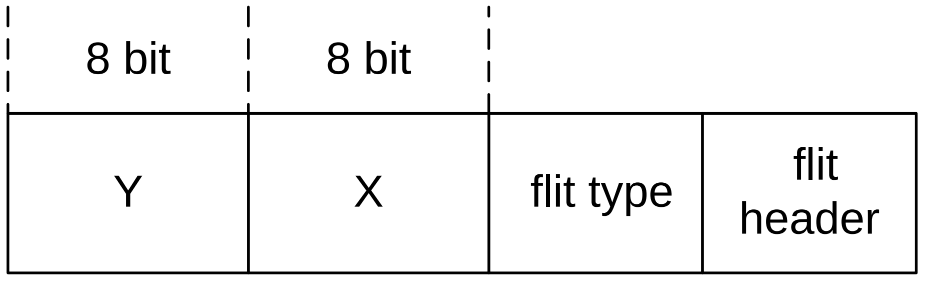


Advanced Processor Technology Group
The School of Computer Science

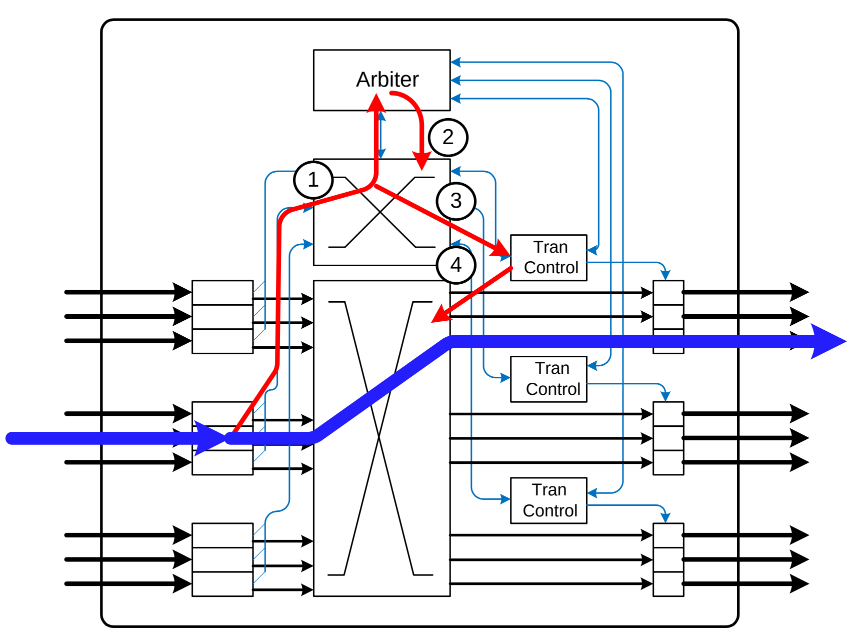
2014/5/13

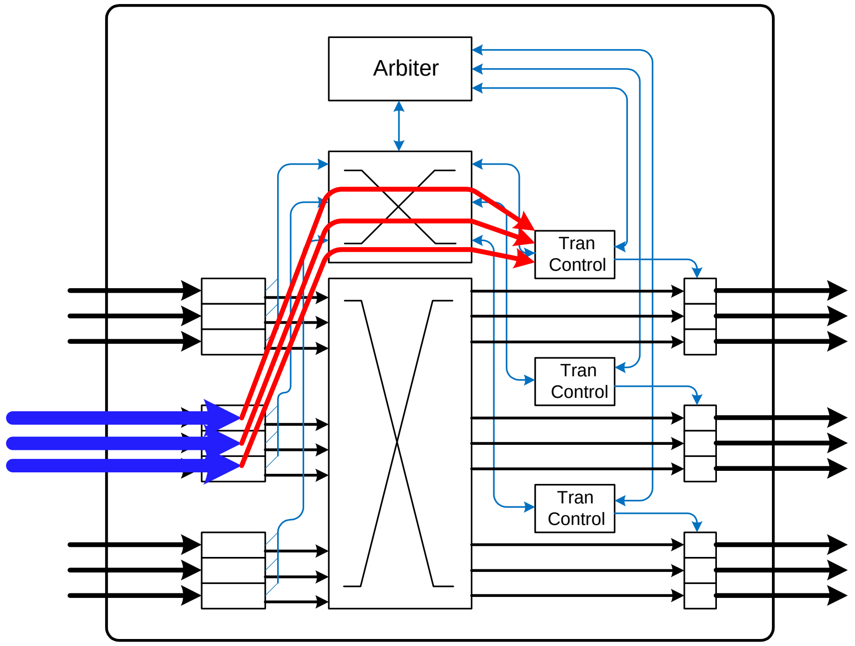
Flit Formats





The Flow Control Procedures





Overview

- A brief review of the *Dynamic Link Allocation* flow control method
- The new simulation platform
- Some simple performance analyses
- An alternative method of the *task request procedure*
- Future schedule

Basic information

– Mesh topology
– Only send XY frames
– Parameter reconfigurable
– Latency is set according to 1-of-4 CHAIN link
– SystemC 2.2.0
– GNU g++
– Makefile
– Batch simulation and automatic result analysis (accepted traffic, latency, loss rate)

Configurable parameters

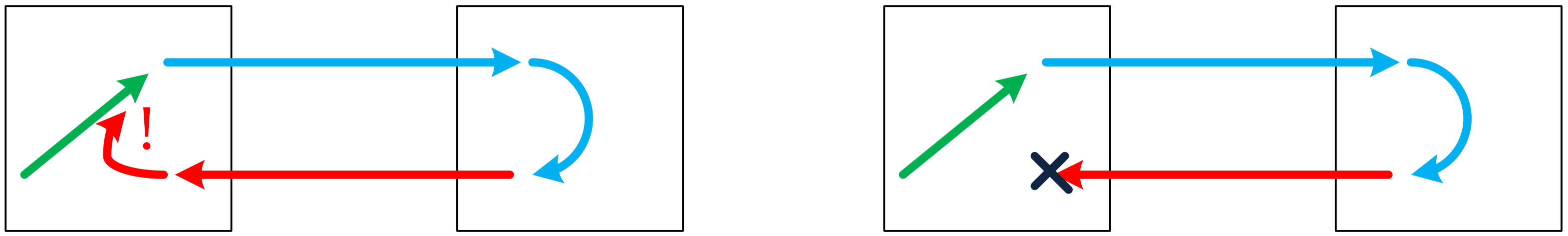
- Dimension (>1)
- Injected traffic (kfps) (>0)
- Channel number (>0)
- Request number (>0)
- Random seed (0 random seed, others seeds)
- Random delay
- Simulation time
- VCD file (generate waveform and debug logs)

Current Problems

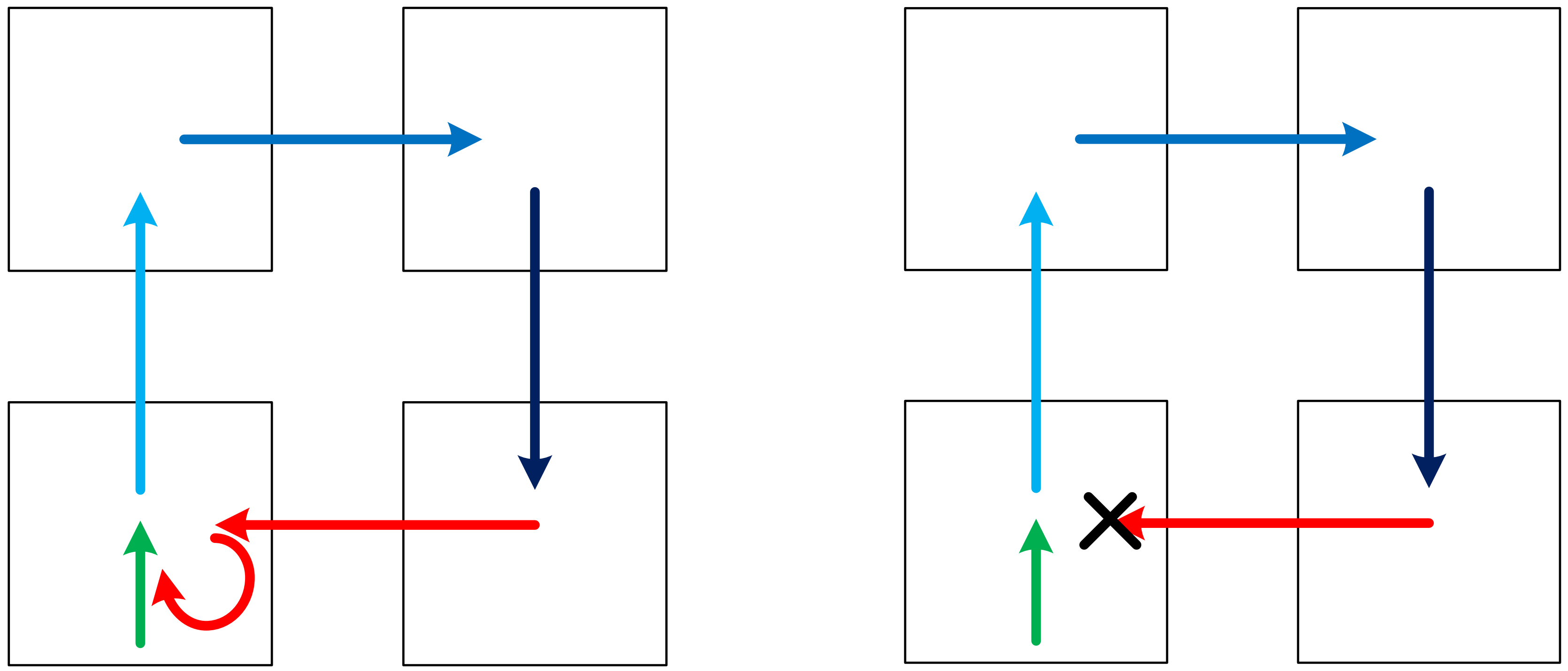
- The router design
  - Multiple request lines sharing one channel will generate deadlocks
    - (still under debugging and modificating)
- The simulation model
  - Slow (possible > 20 min under 4x4 cases)
  - Memory consuming (possible > 2G under some 4x4 cases)

Simulation environment: ADM 2.4GHz 64-bit 4G memory

Deadlock Avoidance 1



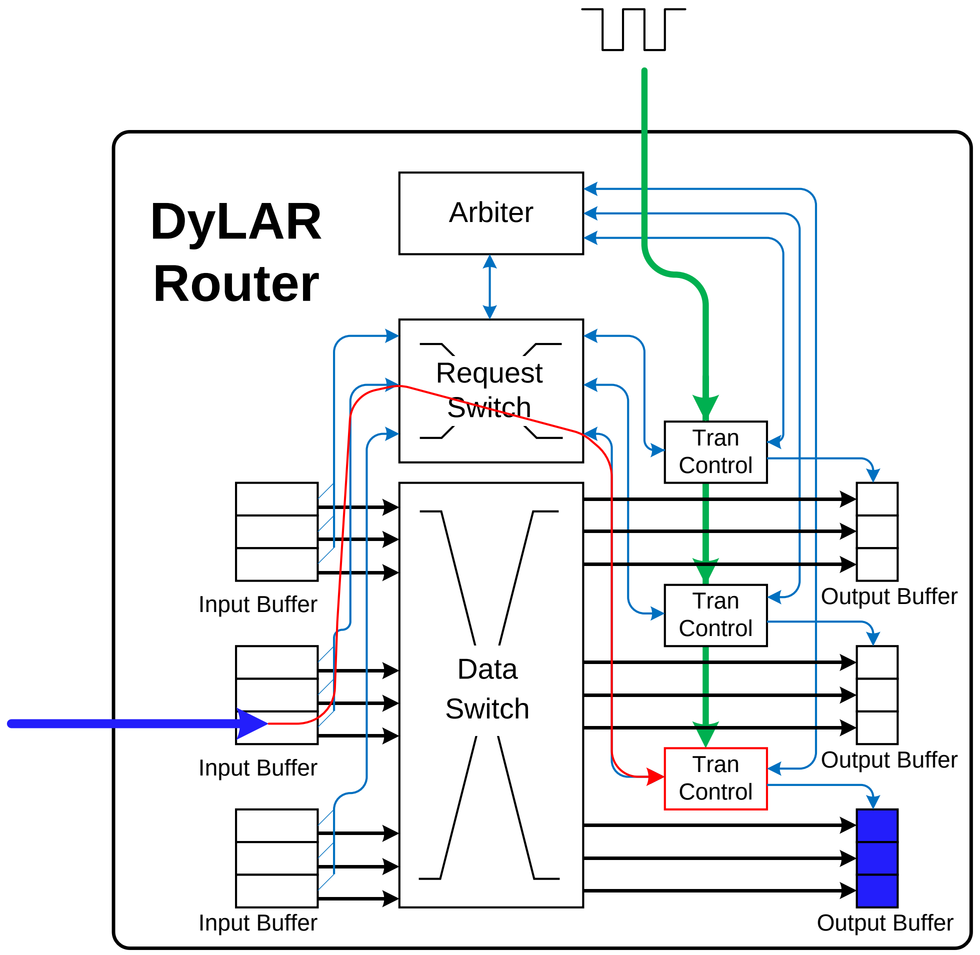
Deadlock Avoidance 2



Advanced Processor Technology Group
The School of Computer Science

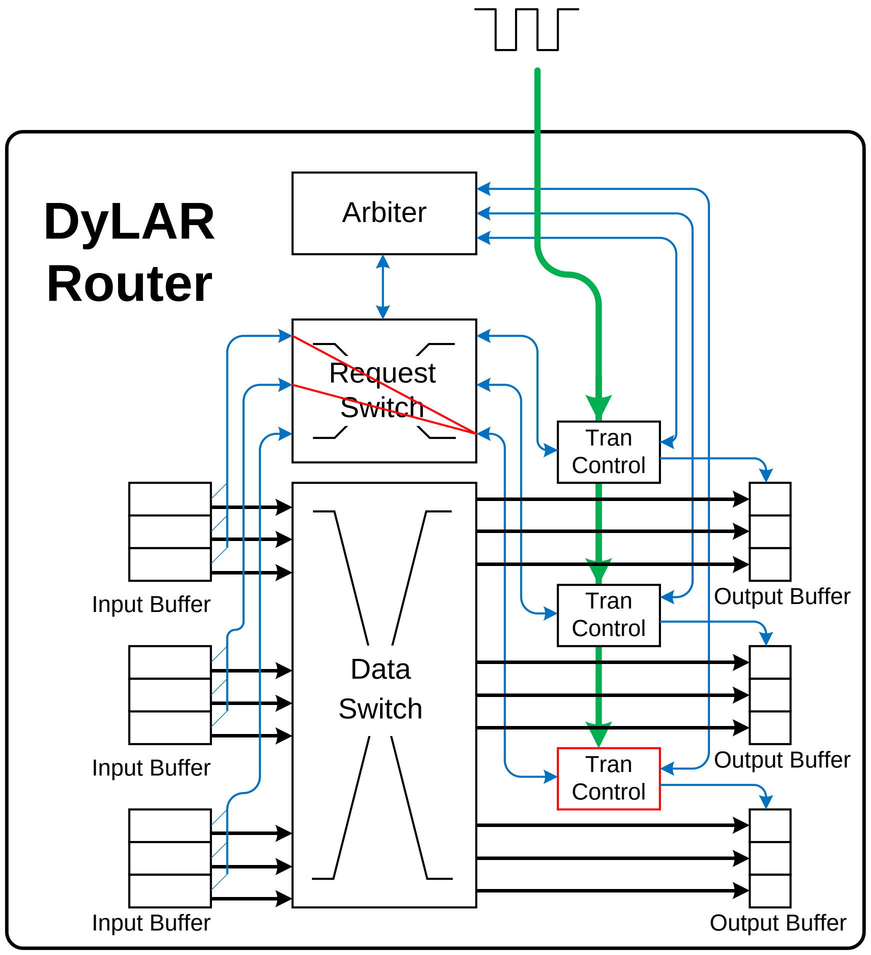
2014/5/13

Deadlock Recovery 1



The University of Manchester

Deadlock Recovery 2



MANCHESTER
1824

The University
of Manchester

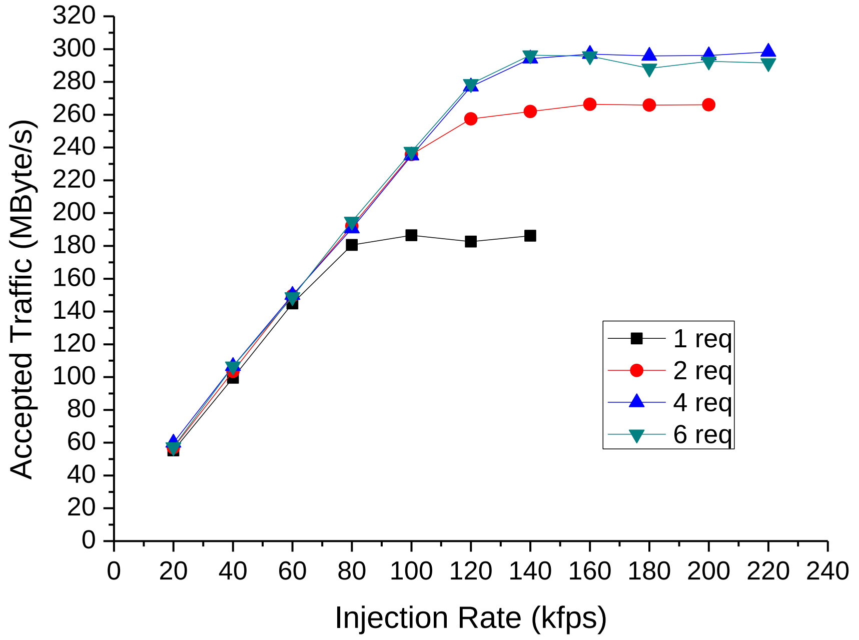
Overview

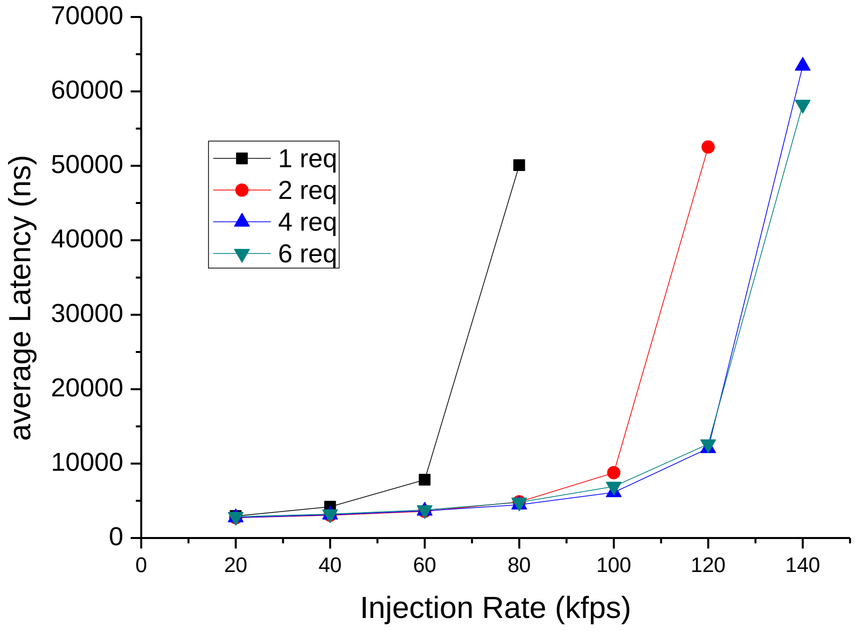
- A brief review of the *Dynamic Link Allocation* flow control method
- The new simulation platform
- Some simple performance analyses
- An alternative method of the *task request procedure*
- Future schedule

Simulation parameters

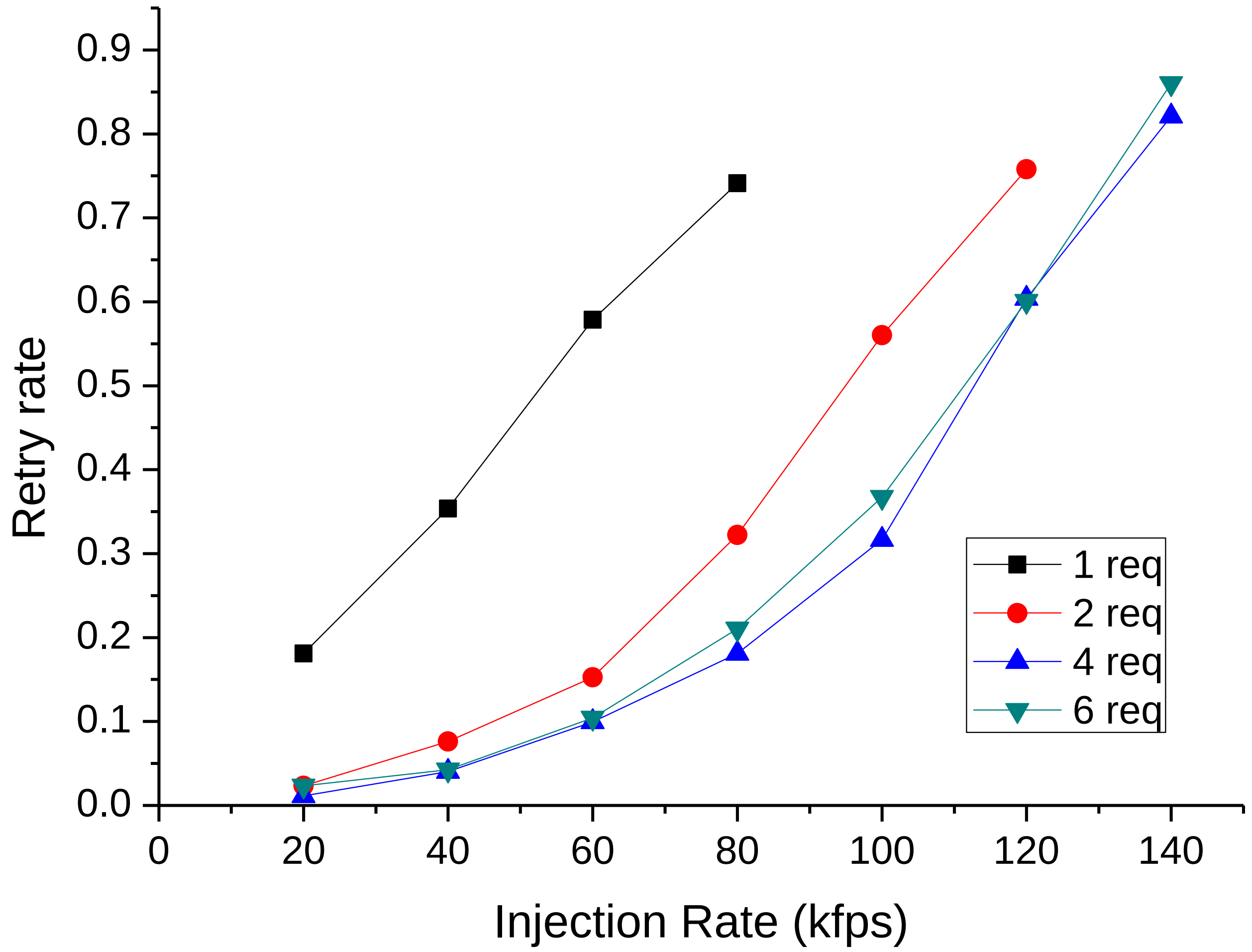
- Dimension 4x4
- Channel 1~3
- Request line 1~8
- Frame injection rate 20~500 kfps
- Random delay and random uniform traffic pattern

1 channel with multiple requests

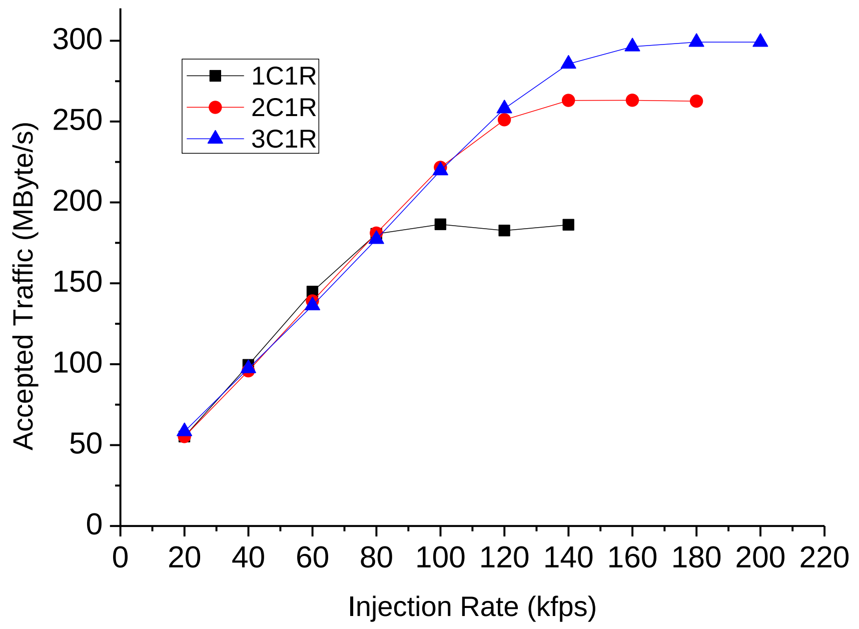


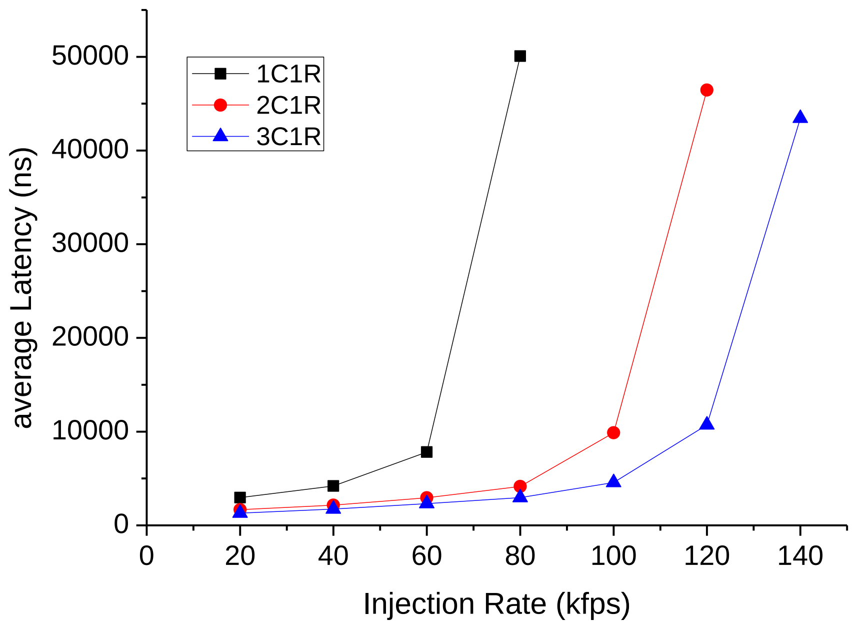


1 channel with multiple requests

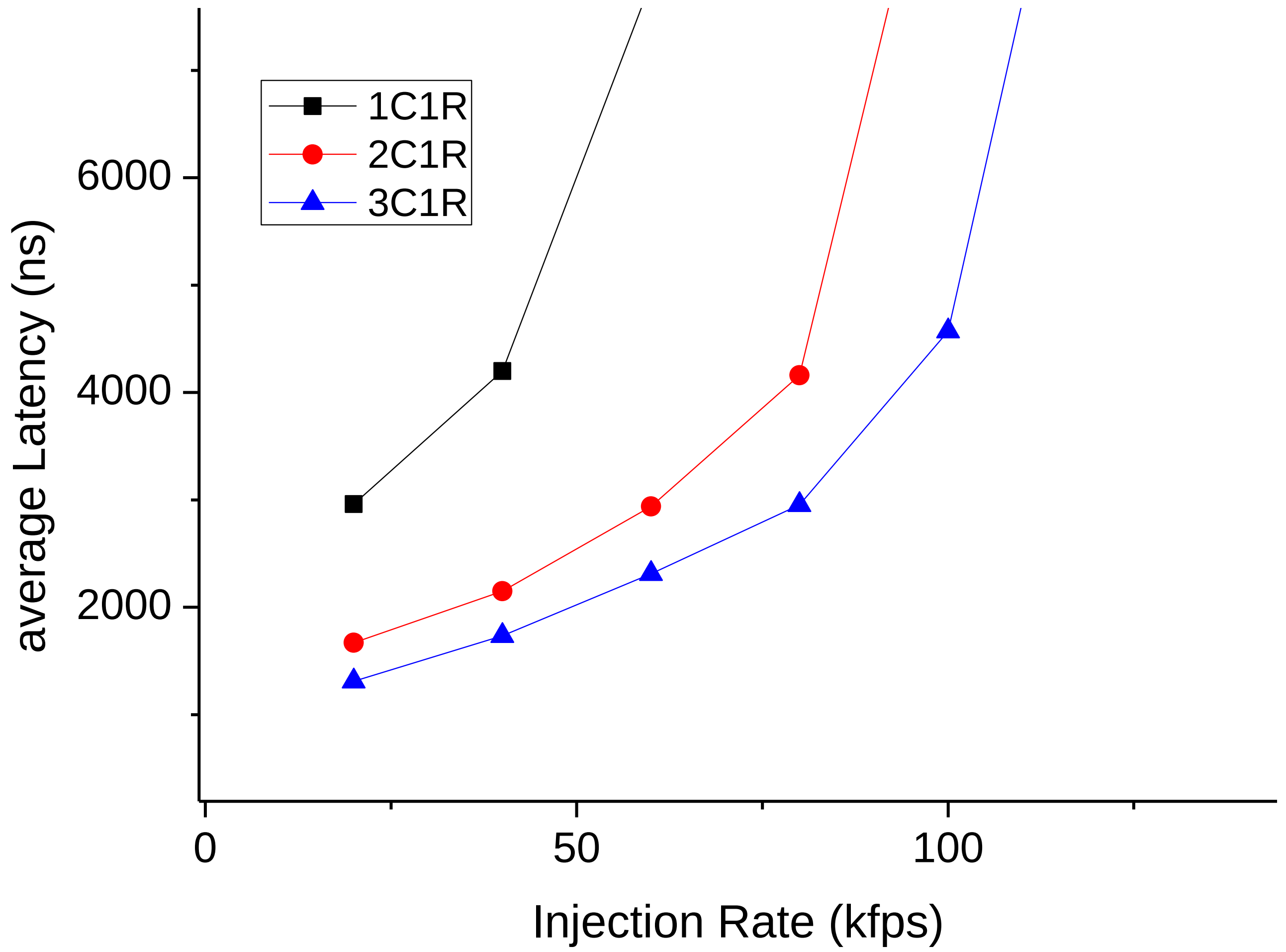


1 request with multiple channels

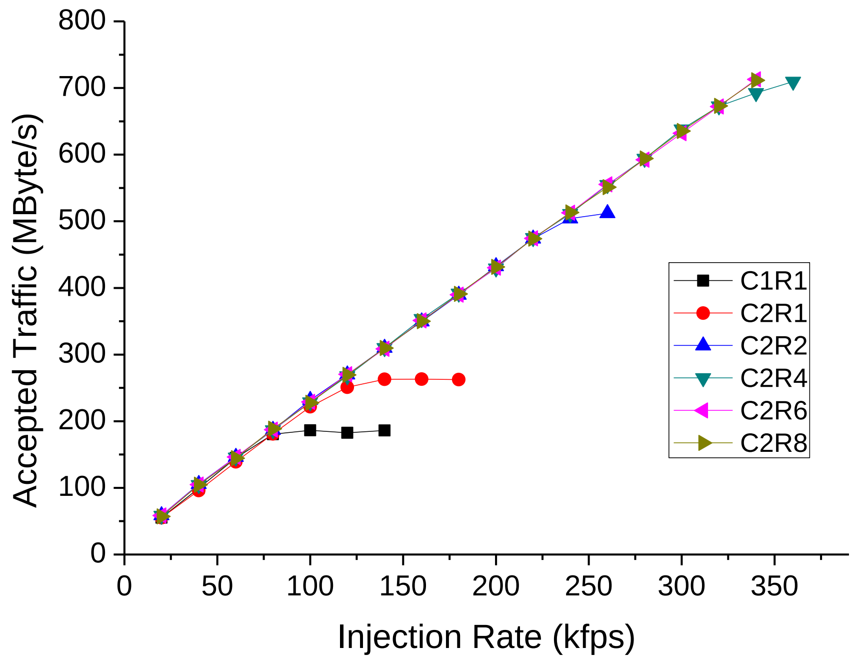


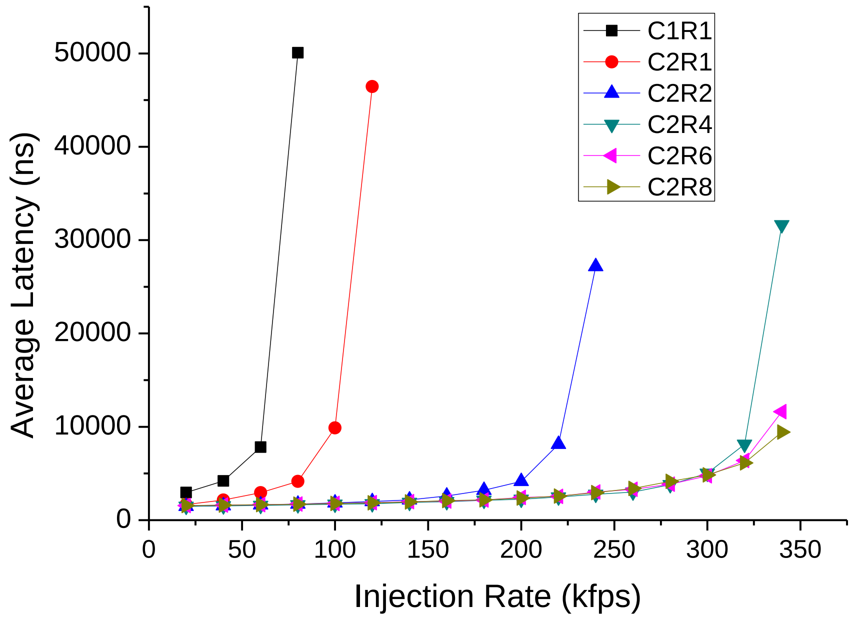


1 request with multiple channels

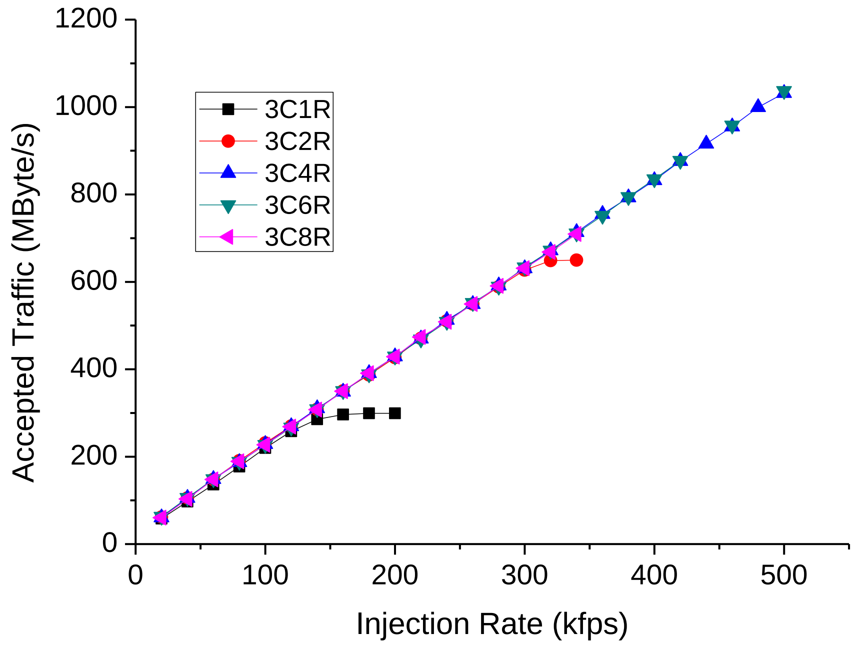


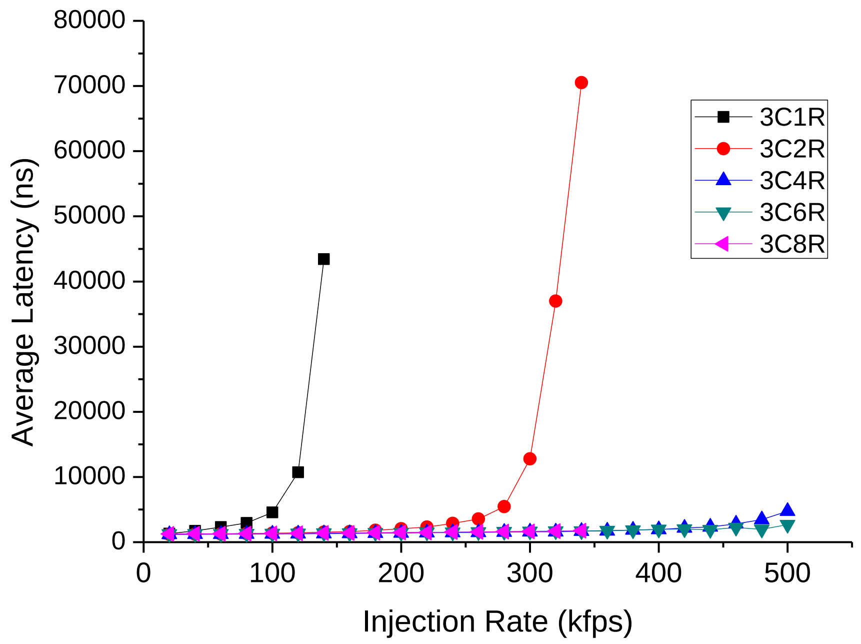
2 channels with multi-requests





3 channels with multi-requests





Throughput

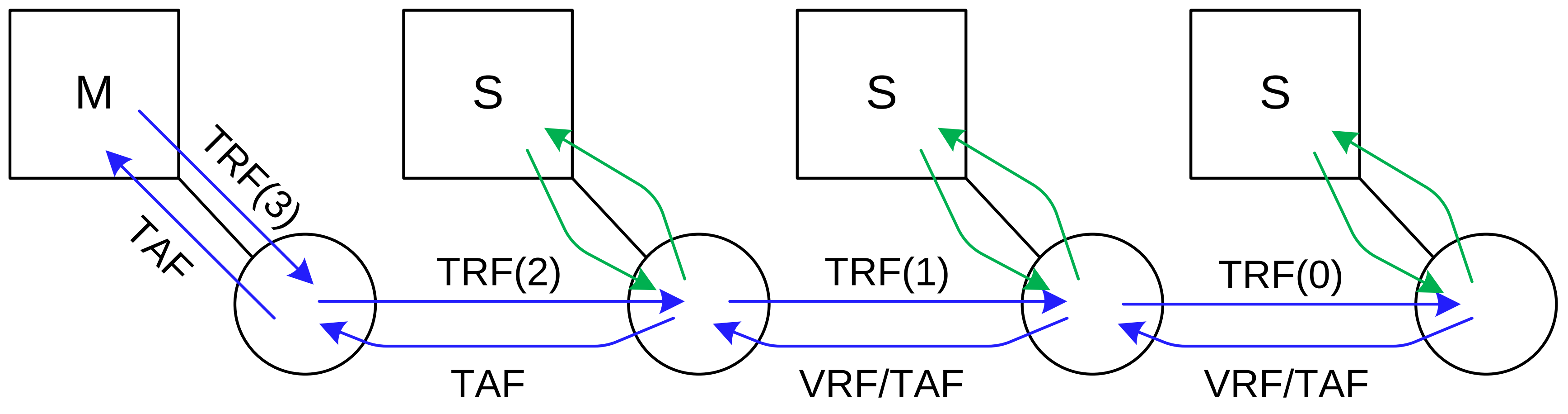
	1 request	2 request	4 request	6 request	8 request
1 channel	186	266	300	300	300
2 channel	265	512	710	>710	>710
3 channel	300	650	>1000	>1000	>1000

Unit: MByte/s

Overview

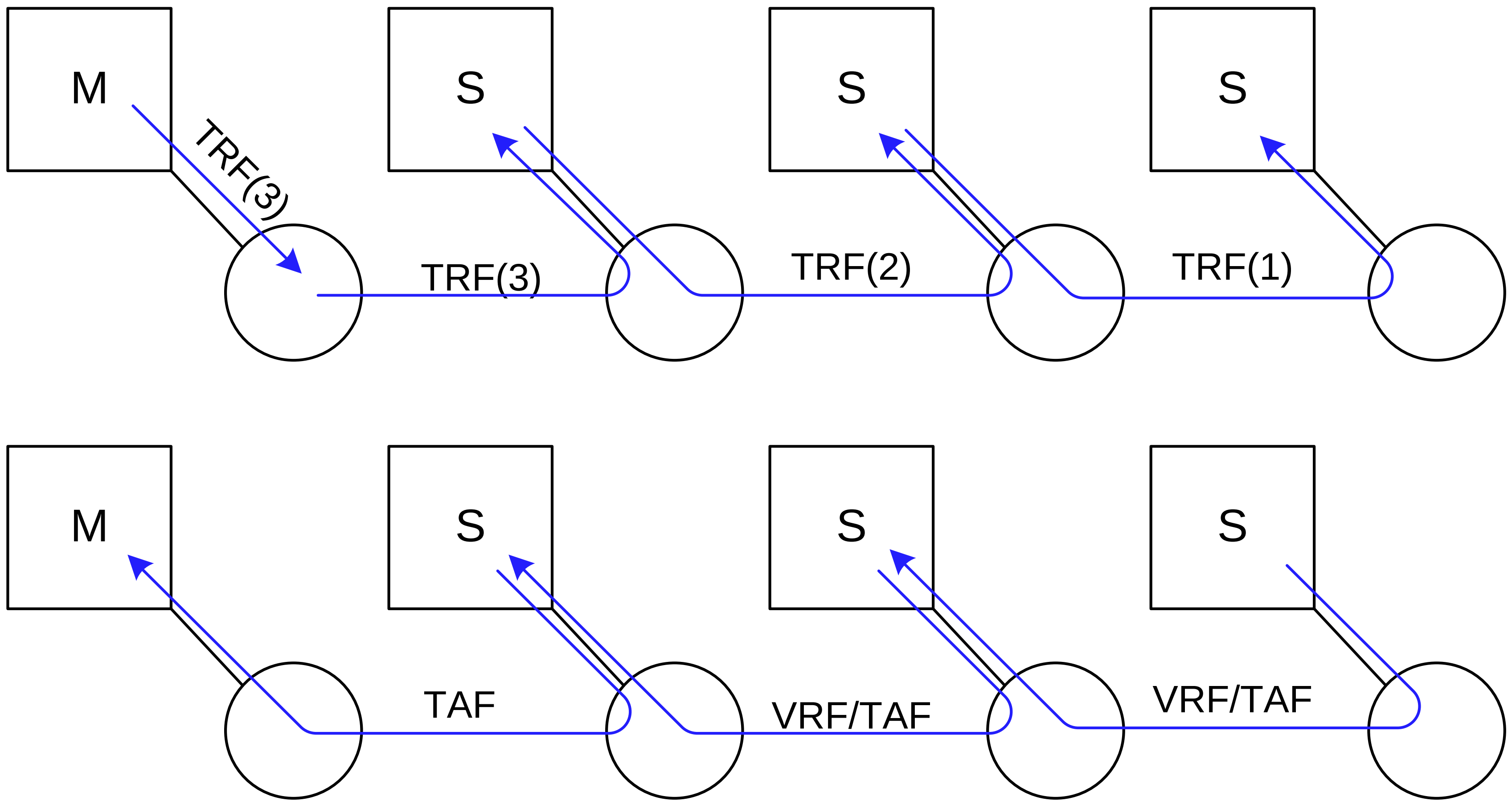
- A brief review of the *Dynamic Link Allocation* flow control method
- The new simulation platform
- Some simple performance analyses
- An alternative method of the *task request procedure*
- Future schedule

The Original Task Request Procedure



TRF task request flit
VRF volunteer request flit
TAF task acknowledge flit

The alternative method



MANCHESTER 1824

The University of Manchester

Comparison of the two methods

- The original TRF
  - Need counters to calcuate life_time
  - Remember state for every TRF
  - Special communication with NA
  - Wait for the whole flit
  - One request line per TRF
- The alternative
  - Move counters to NA
  - States will be recorded by NA and only 1 state machine is enough
  - Directly send flit to NA
  - Send directly after the flit_type field
  - Two request lines per TRF

Overview

- A brief review of the *Dynamic Link Allocation* flow control method
- The new simulation platform
- Some simple performance analyses
- An alternative method of the *task request procedure*
- Future schedule

MANCHESTER 1824

The University of Manchester

Schedule

- The simulation model is still under debugging
- Build the hardware model according to the SystemC model
- Try to speed up the simulation model and reduce the memory required

Thank you!

Questions?