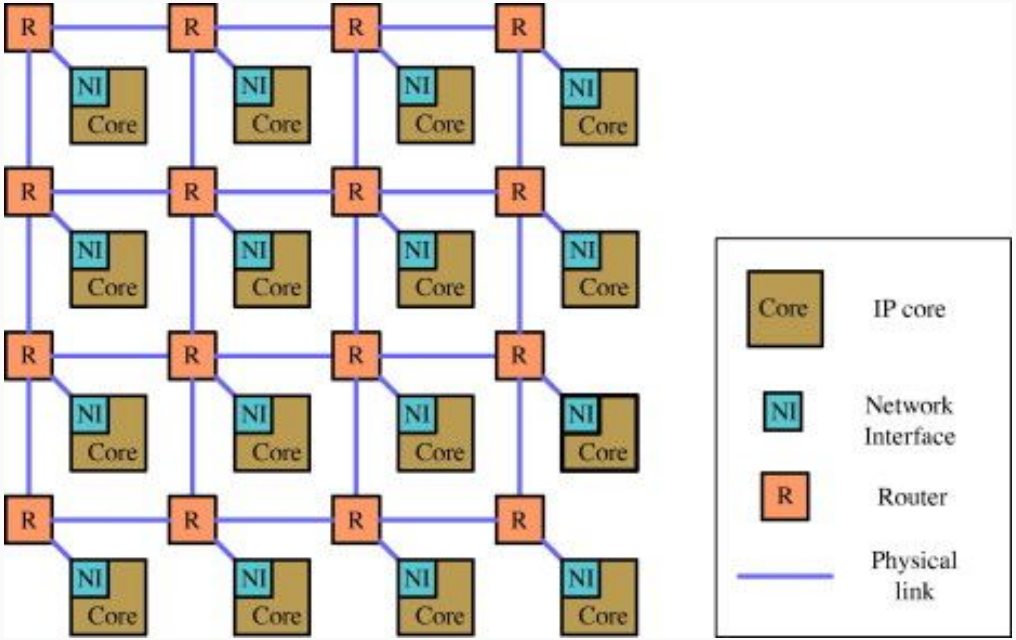
PANE: Pluggable Asynchronous Network-on-Chip Simulator

Sarabjeet Singh • 24.04.2017
Under guidance of,
Prof. Joycee Mekie

Network-on-Chip (NoC)



- System level NoC parameters such as packet latencies, throughput.
- Existing Synchronous NoC tools such as BookSim2[1], Noxim.
- Heterogeneous Multi Processors (HMP), such as Nvidia Xavier, Parker and Tegra XI, Samsung Exynos 8895, where communication across clock domains is required. -> unidentical routers and links!

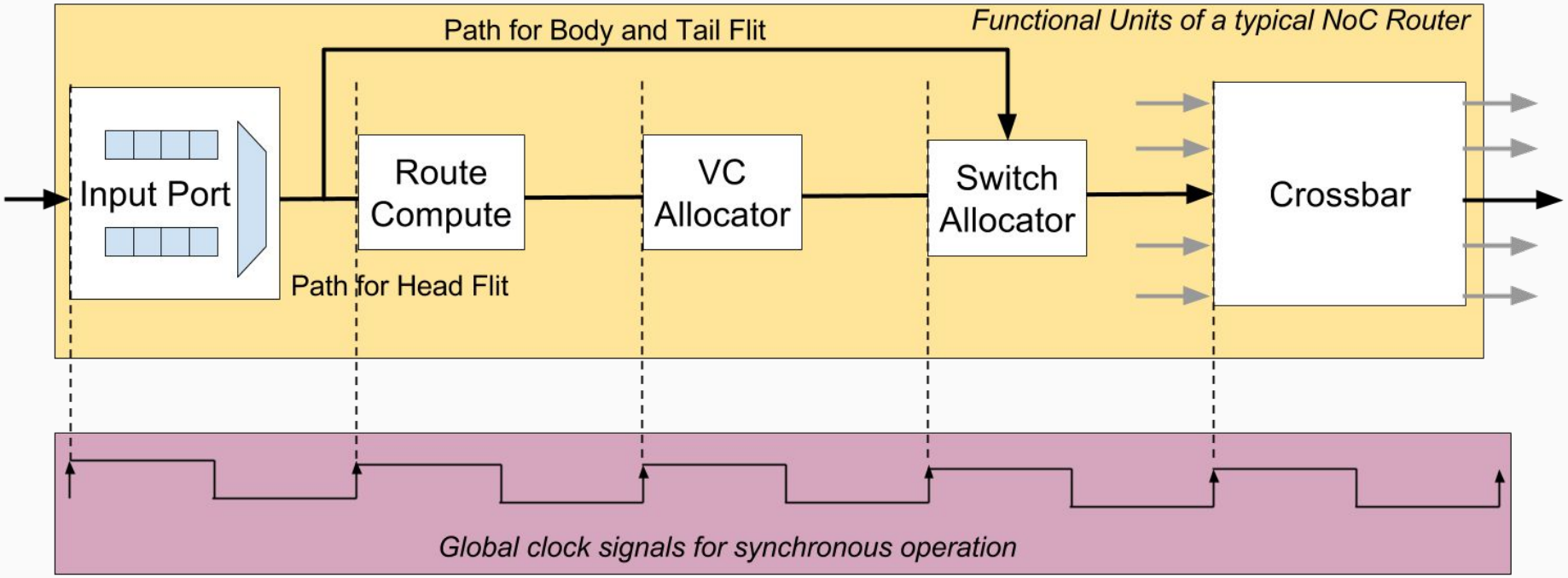
Therefore, PANE!

Pluggable Asynchronous Network-on-Chip Simulator

an event-driven asynchronous NoC simulator

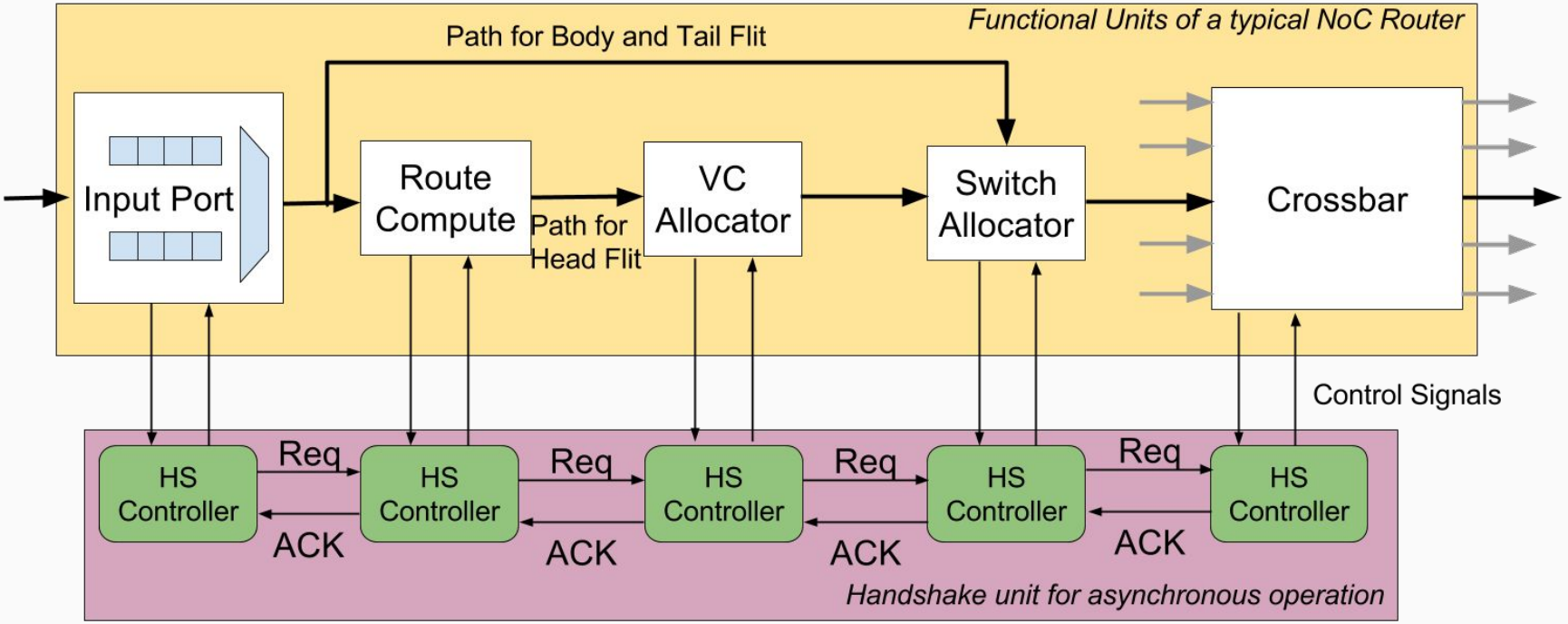
- ➔ UNIX compatible tool
- ➔ Model a variety of NoC designs:
  - ◆ Different topologies
  - ◆ Routing algorithms, allocation and arbitration mechanisms
  - ◆ Flit widths and buffering schemes
  - ◆ Real benchmark applications and synthetic traffic patterns.
- ➔ Complete system level analysis of synchronous, asynchronous and heterogeneous NoC architectures.

Router Design: Synchronous Circuits



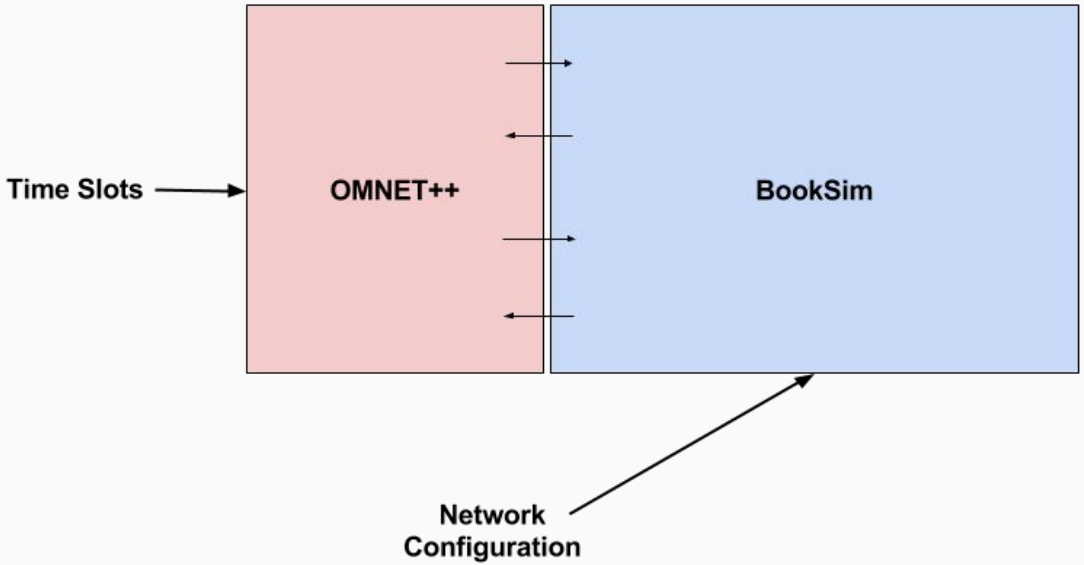

```
if( CLOCK == TRUE)
    SendNext();
```

Router Design: Asynchronous Circuits




```
if( ( TimeSlot == TRUE ) && ( Ack == TRUE ) )
        SendNext();
```

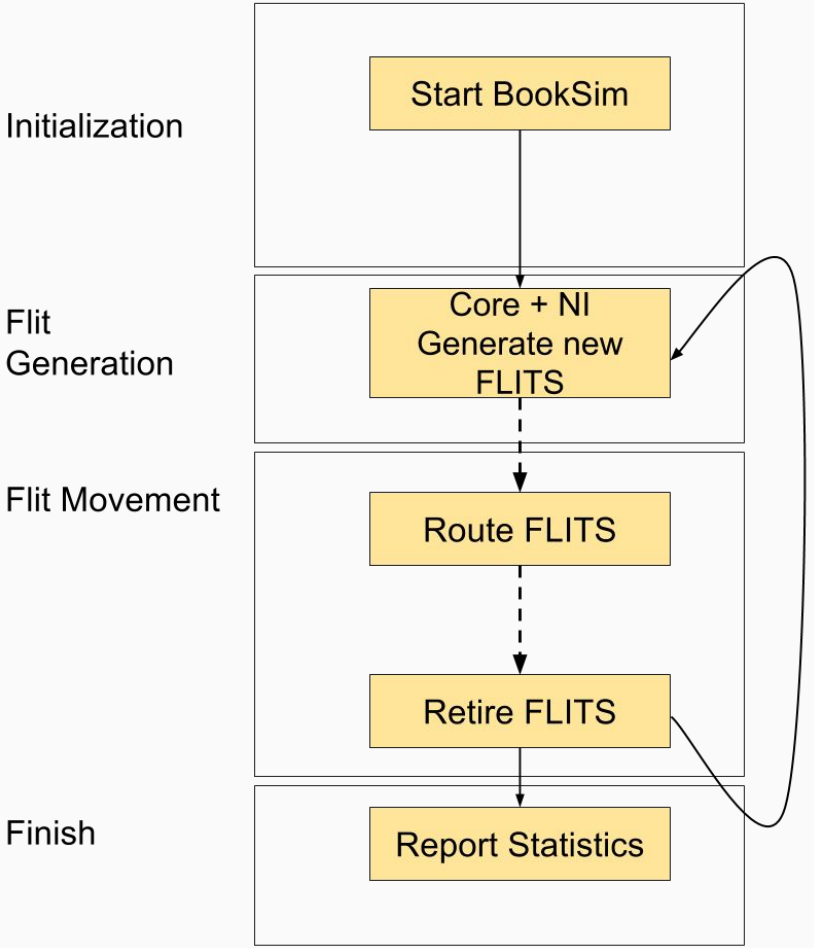
PANE



BookSim2[1]

A Cycle-Accurate Interconnection Network Simulator

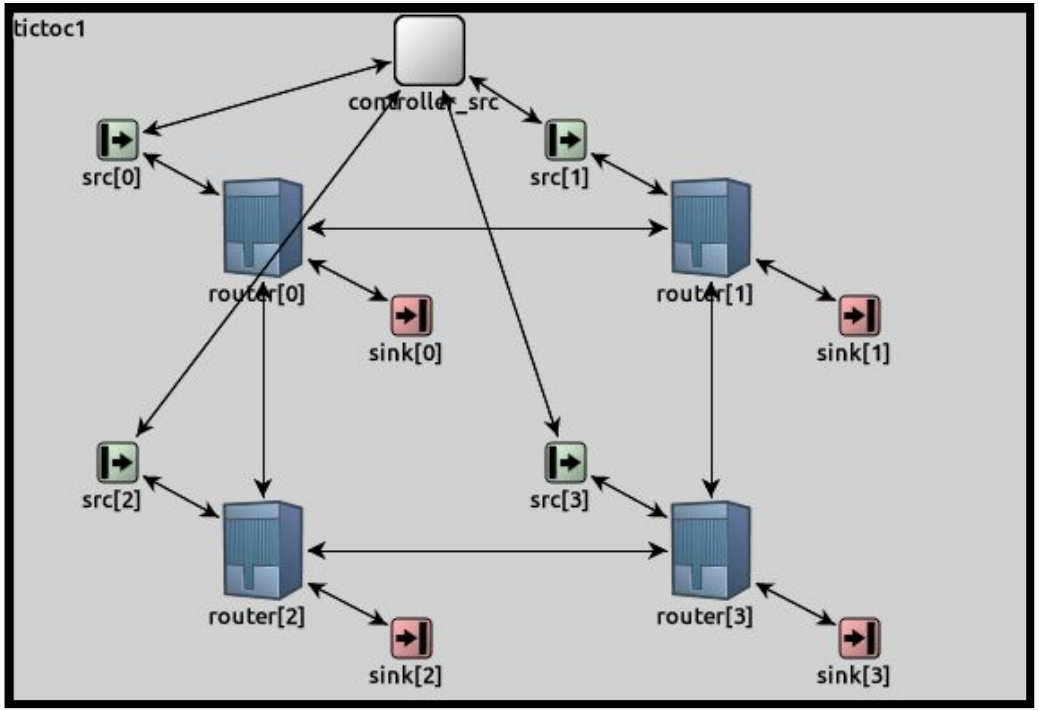
- C++ based synchronous NoC Simulator
- BookSim2[1] gives a detailed description of the routers and the interconnect behavior for the NoC designers to test and validate their NoCs.
- However, BookSim2[1] is designed to simulate only homogeneous NoCs where all the routers operate at the same clock frequency and does not support asynchronous and heterogeneous NoC simulation.

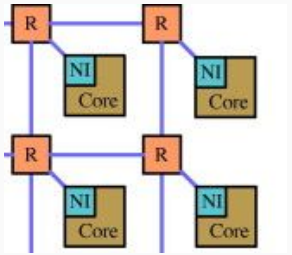


OMNeT++[2]

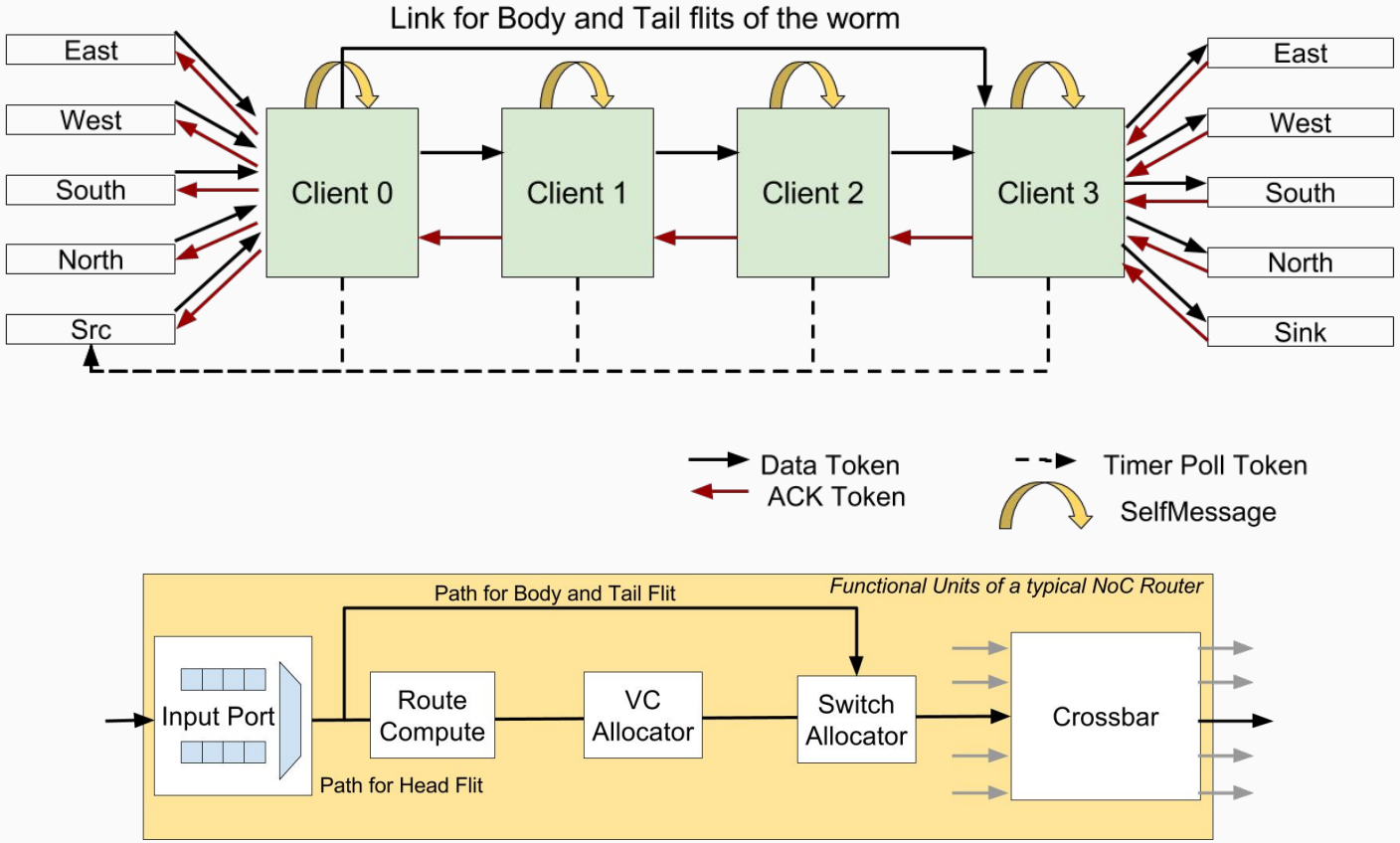
Discrete Event Simulator

- ➔ C++ framework to design and simulate network designs.
- ➔ Supports event-driven simulation!
- ➔ Network topologies can be easily described in OMNeT++
- ➔ Additionally it provides network animation, which helps in network visualization, debugging and traceability.
- ➔ OMNeT++ is available as open-source, free for non-profit use and is used by a fairly large community of users for research and educational purposes.





Block Diagram of OMNeT++ Router

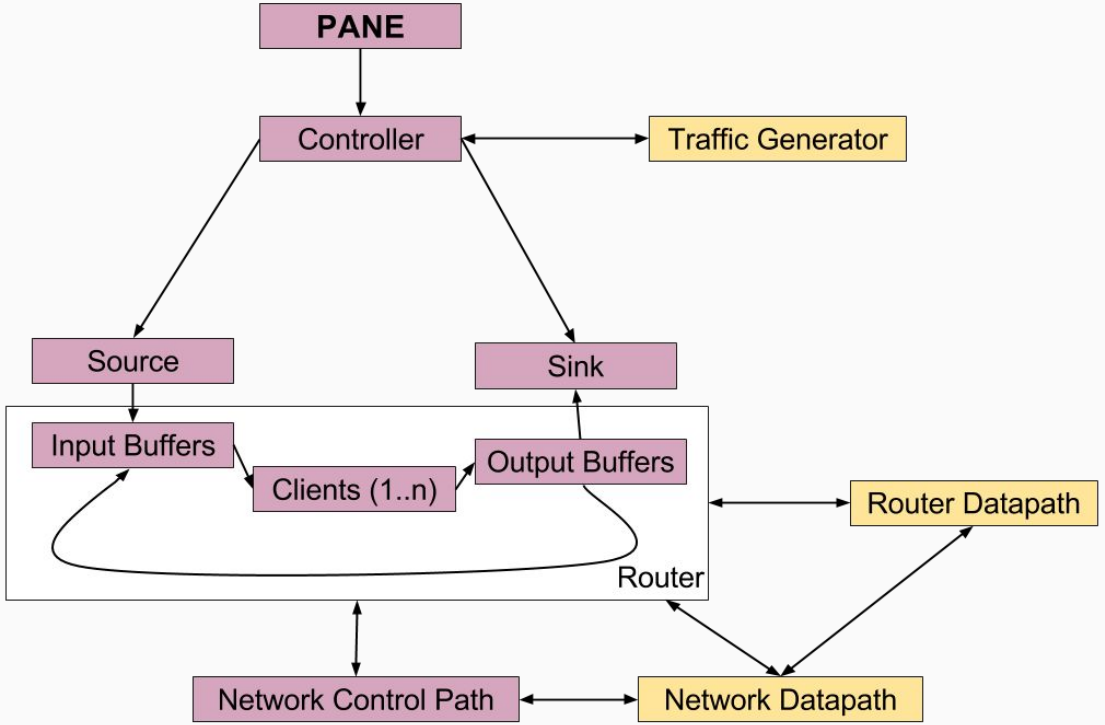


MY WORK

Building an OMNeT++[2] plug-in to extend BookSim2[1] for heterogeneous NoCs

- ➔ Building an OMNeT++ model of an NoC
- ➔ Establishing communication between BookSim and OMNeT++[2]

Connection graph of modules in PANE

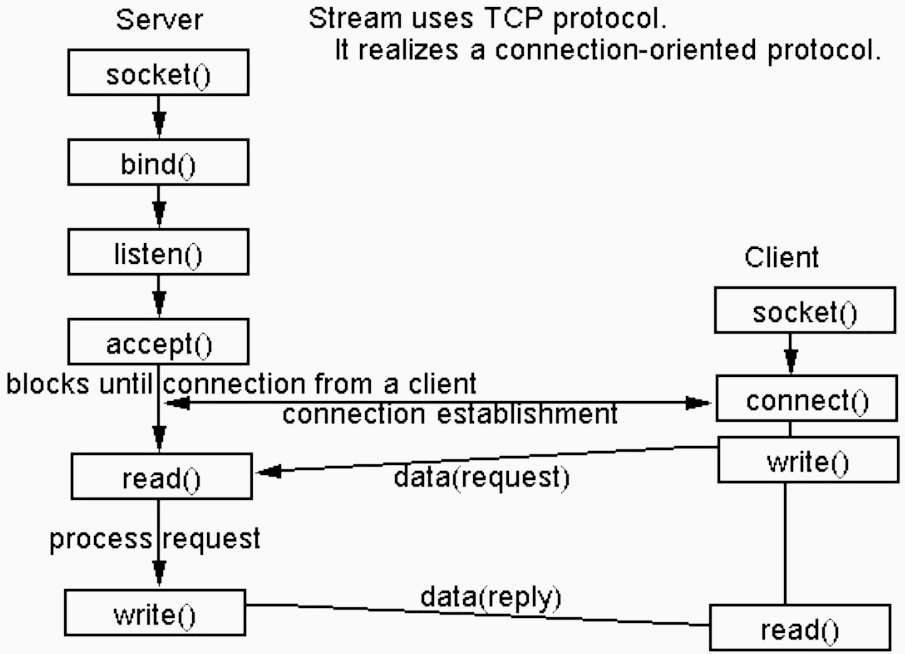


IPC - sockets[3]

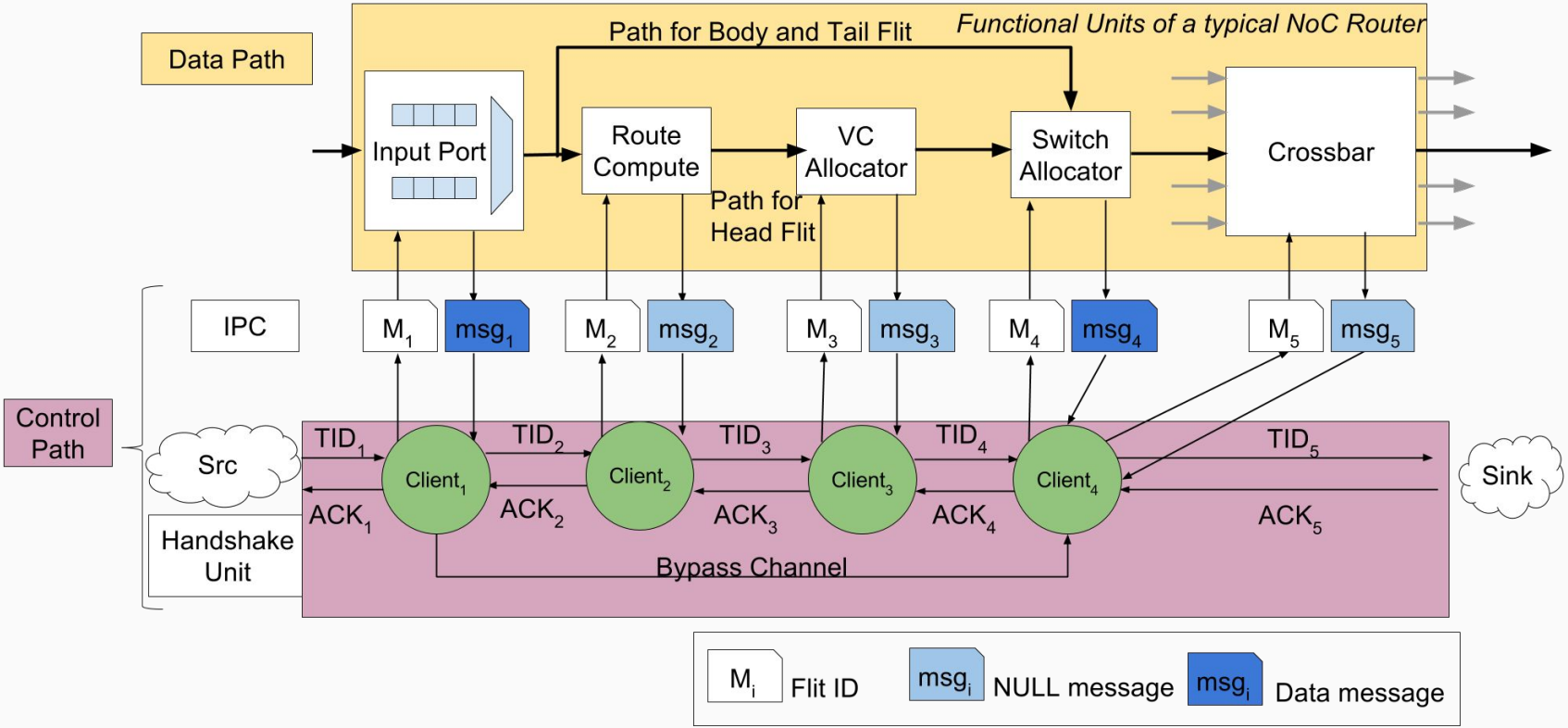
Sockets provide point-to-point, two-way communication between two processes.

UNIX domain sockets allow IPC among two different processes running on the same machine.

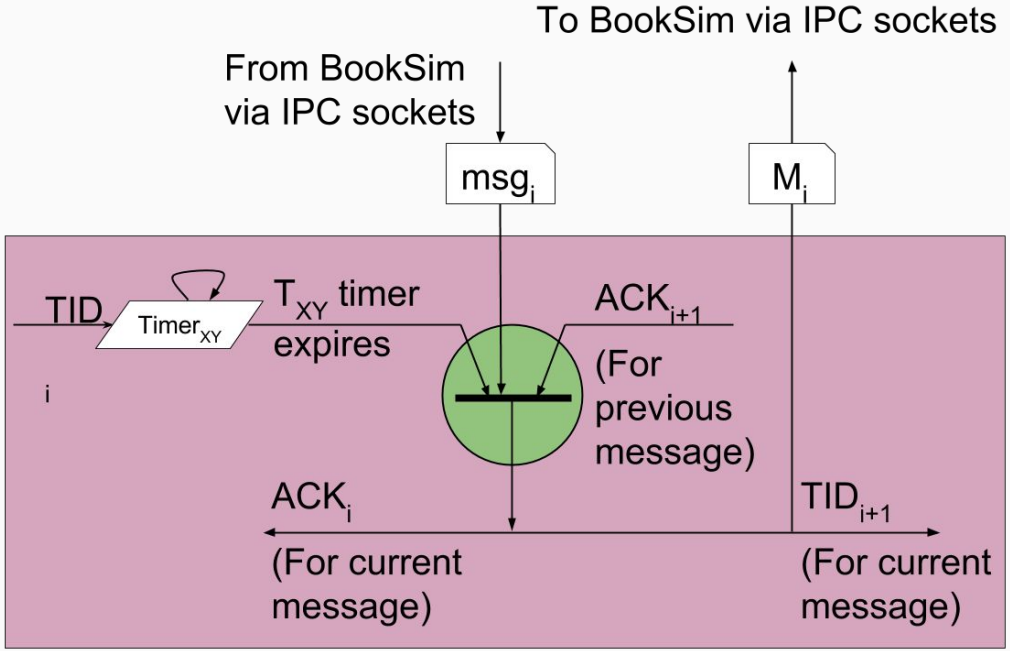
No ancestor process required

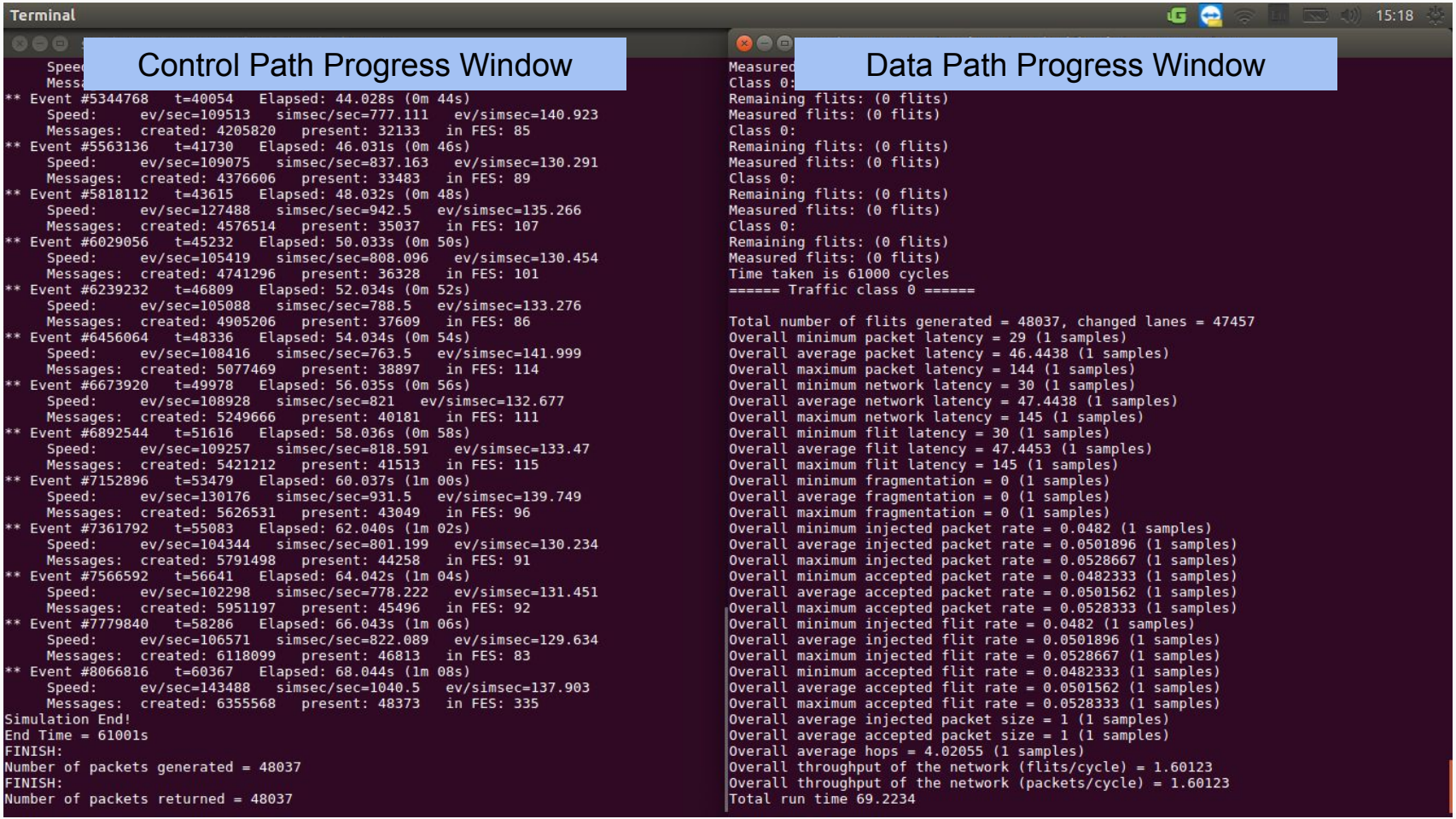


Schematic representation of asynchronous router operation in PANE



Schematic of a PANE Client





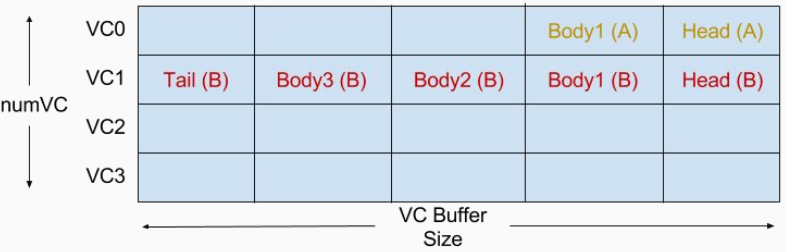
Snippet of PANE running

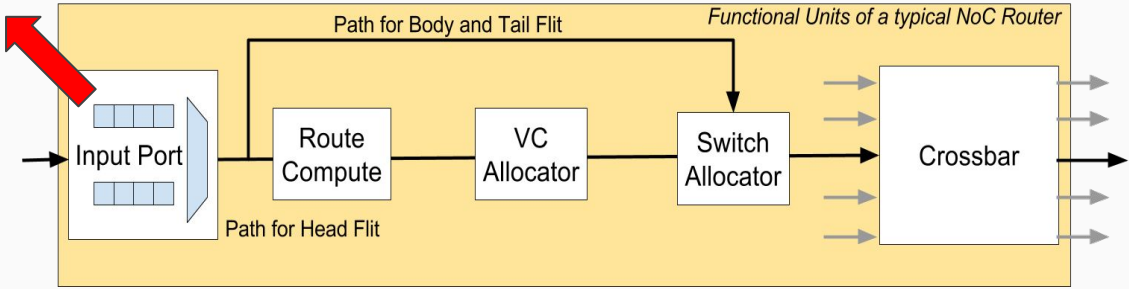
Fixing Memory Leaks

- Memory consumption of 4GB and increasing to a saturation point of 50MB!
- Exhaustive search through Valgrind
- Bug: Self-msgs were relocated to new chunk of memory before being deleted

Optimization: Number of sockets

- PANE Version 1 had 8X8X5X5 = 1600 open sockets! *(for a 8X8 Mesh network)*
- Different Buffer Organization





- Initially, no of sockets = 8X8X5X5X4 = 6400!! *(for a 8X8 Mesh network, with 4 Virtual Channels)*

Optimization: Number of sockets

- Initially, no of sockets = 8X8X5X5X4 = 6400!! *(for a 8X8 Mesh network, with 4 Virtual Channels)*

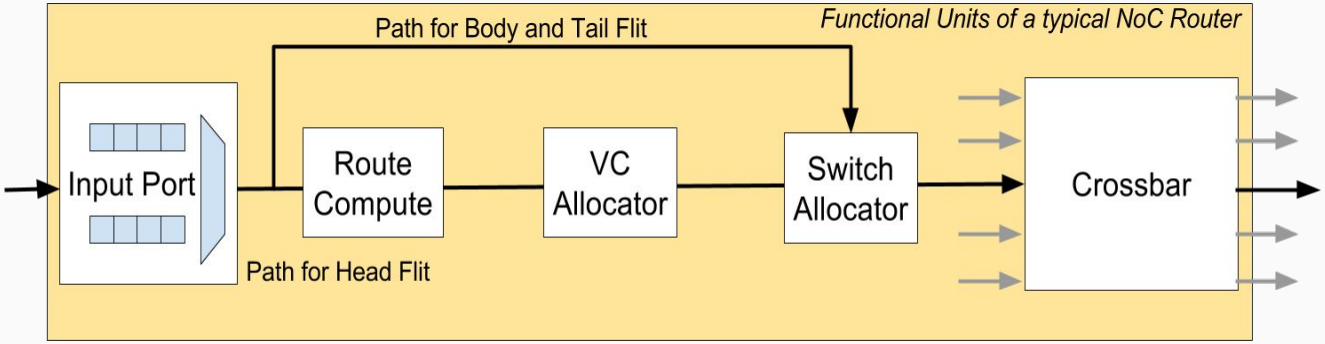
- 1 Data over 1 socket per cycle -----> Multiple Data over 1 socket per cycle.

Problem: UNIX Domain Sockets map according to activity and not number of transfers done.
Solution: Concatenation of '*' after multiple data and making VC# as an attribute of the msg.

- Reduced to 8X8X5X4 = 1280 sockets! *(for a 8X8 Mesh network, with any number of Virtual Channels)*

Optimization: Simulation Time

- Extension to multiple flit size
- Routing only the head of the packet.



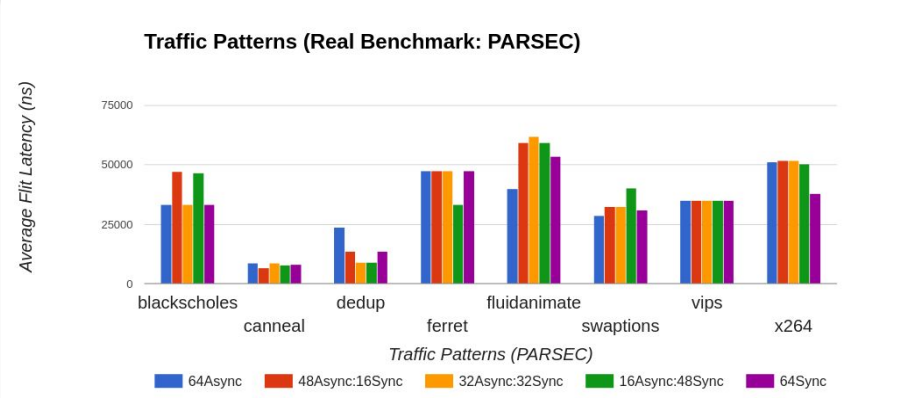
Scripts

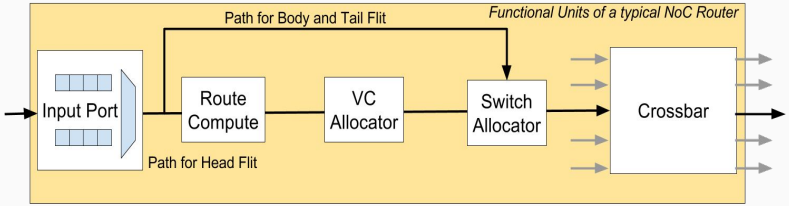
For changing configurations, running and collecting statistics of:

- Traffic Pattern: Synthetic and Real
- Routing Function
- Packet Size
- Injection Rate
- Network Size: Sockets
- Connection(ned) file: Topology and network size

Miscellaneous

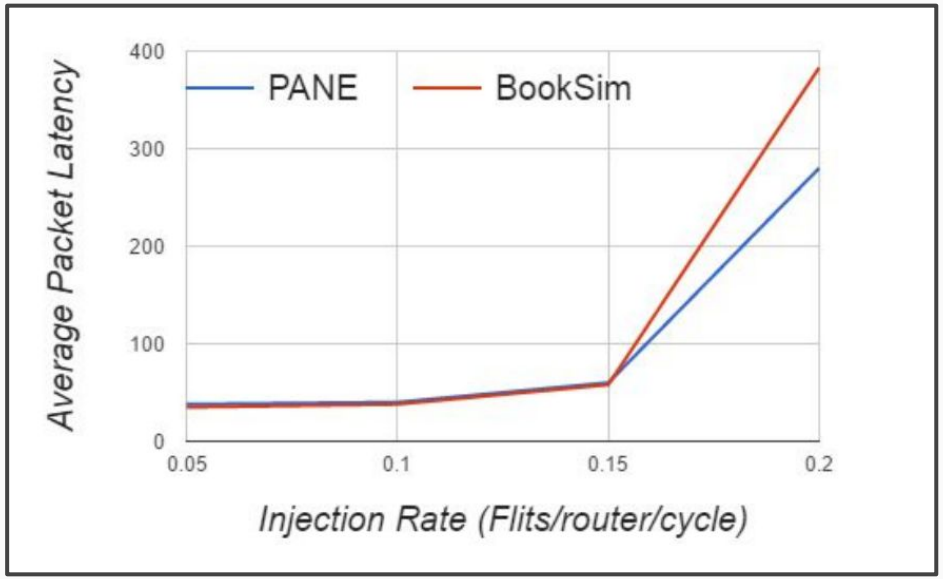
- Addition of Configurable Heterogeneous Network
- BookSim Synchronous Behaviour Assumptions - Flush all data each cycle!
- PANE runs even if there are no packets in the network.
- Header files for all socket and miscellaneous functions





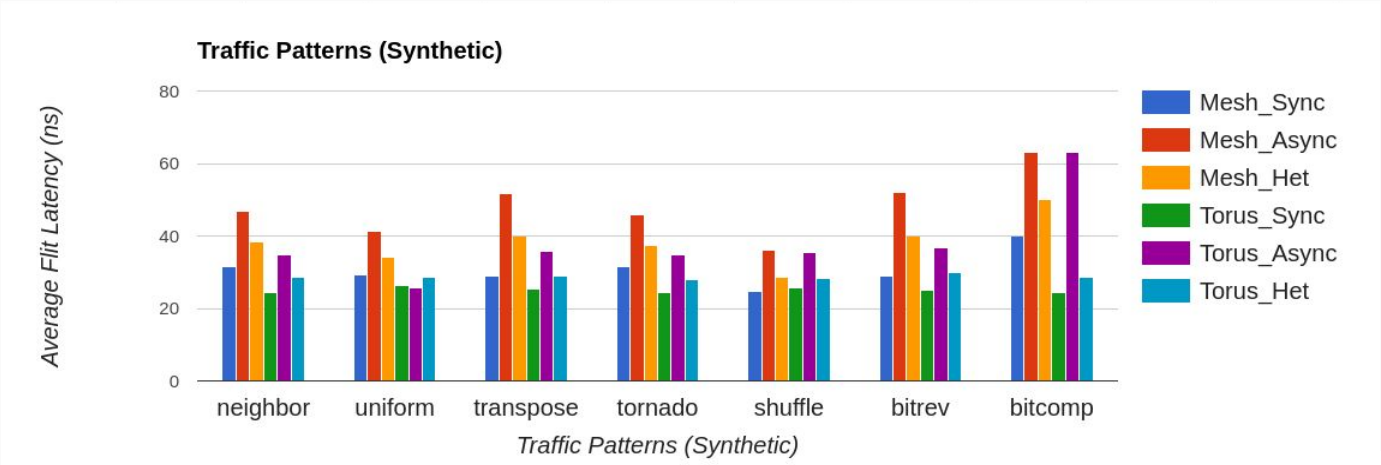
Validating PANE With Synchronous NoC Simulator

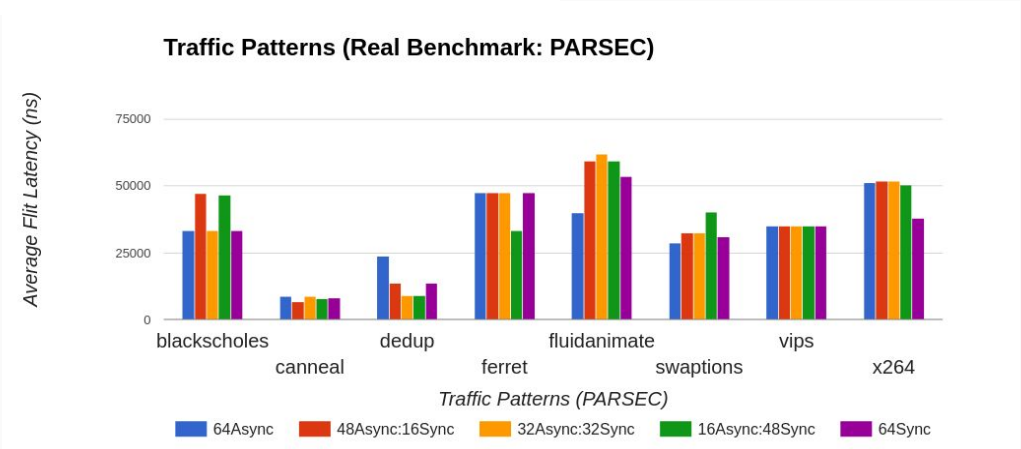
BookSim2[1] can be used as a reliable reference for validation. The PANE implementation, for the validation experiment, is such that each PANE client takes unit time to complete its action. In other words, PANE is configured to simulate a synchronous NoC. Since this is the same as simulating a NoC in BookSim2[1], the results obtained after both the simulations should match.



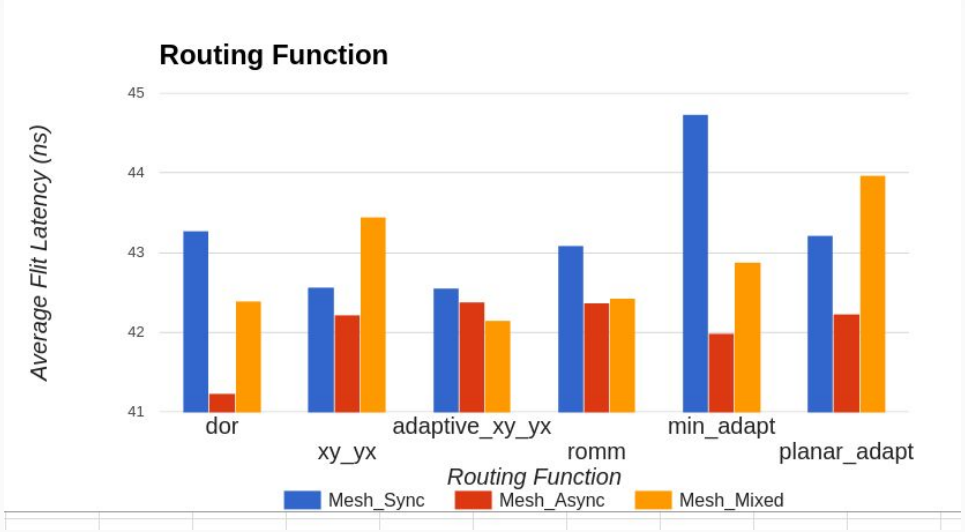
PLOTS

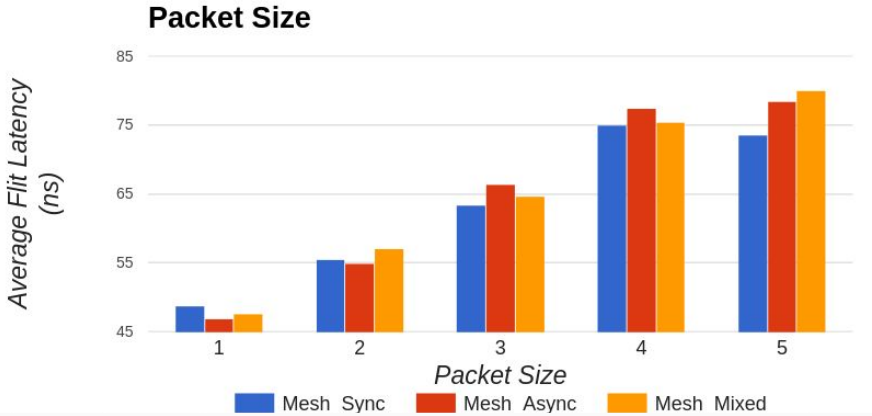
Traffic Pattern: Synthetic and PARSEC



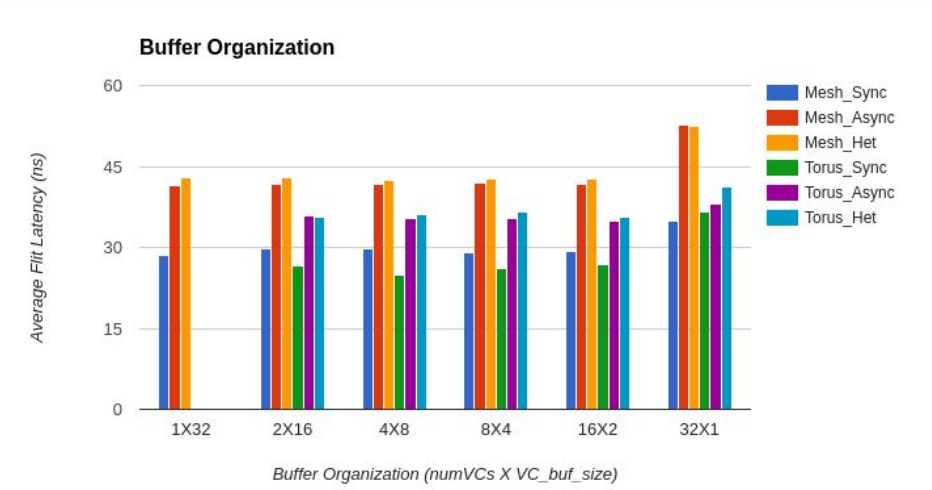


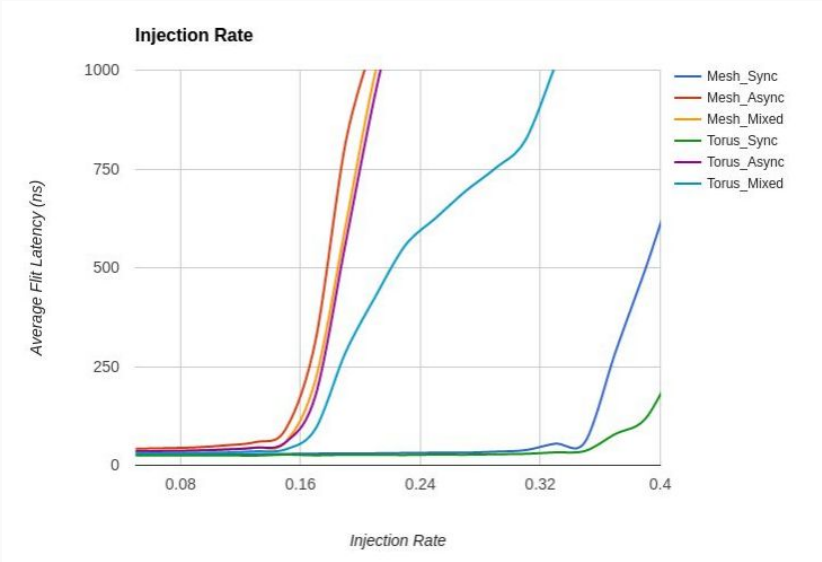
Routing Function & Packet Size





Buffer Organization & Injection Rate





Next steps

Progress Area 1

Reduce the number of Sockets by concatenation of multiple data over single socket!

Progress Area 2

Reduce the Simulation Time.

Progress Area 3

Develop a GUI and automation scripts for configuring and collecting statistics.

DEMO

Thank You!

References

1. Nan Jiang, Daniel U. Becker, George Michelogiannakis, James D. Balfour, Brian Towles, David E. Shaw, John Kim, and William J. Dally. 2013. A Detailed and Flexible Cycle-Accurate Network-on-Chip Simulator. In ISPASS.
2. Andras Varga and Rudolf Hornig. 2008. An Overview of the OMNeT++ Simulation Environment. In Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops (Simutools '08).
3. Stevens, W. Richard, Bill Fenner, and Andrew M. Rudoff. UNIX network programming. Vol. 1. Addison-Wesley Professional, 2004.