www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 5 May 2015, Page No. 12053-12057

A Traffic Model for Routing Logic in Network On Chip

C.G. Joy Merline[1], R.S. Niveditha[2]

[1]Associate Professor, ECE Department, Madha Engineering College, Chennai.
[2]P.G. Student, VLSI Design, Madha Engineering College, Chennai.

[1]joymerline@gmail.com
[2]nive.prety@gmail.com

***Abstract*- The increased demand for on-chip communication bandwidth as a result of the multi-core trend has made packet-switched networks-on-chip (NoCs) a more compelling choice for the communication backbone in next-generation systems. However, NoC designs have many issues like power, area, and performance trade-offs in topology, buffer sizes, routing algorithms and flow control mechanisms—hence the study of new NoC designs can be very time intensive. The effectiveness of the NoC system can be found by hardware implementation. For this a traffic generator can be designed and the data flow in the NoC can be checked. The Traffic generator with random number generator is used for traffic creation and Bernoulli process based traffic injection is used to inject congestion in the NoC. The TG is also capable of generating error free traffic using it's analyser unit.**

Keywords- NoC(Network-on-chip), packet switching, routing, TG(traffic generator), Bernoulli process, analyser unit.

I. INTRODUCTION

With network performance highly dependent on network traffic, traffic models are critically needed for a thorough understanding of the vast design space of network architectures, protocols and implementations. As a result, there has been extensive prior research into traffic models for diverse networks ranging from the Internet [9, 24], Ethernet [14, 26], wireless LANs [1] to shared-memory-multiprocessor (SMP) networks [4,7,35]. These models provide critical insights into the traffic behavior of these networks and have been used effectively in evaluating existing and new network designs. A new class of networks where the entire network resides on a single chip has emerged in recent years and is gaining in importance [3, 6, 23, 31]. These on-chip networks1 re- place traditional bus-based and crossbar-based interconnects, providing a scalable communication fabric between multiple processor cores on a single chip where inter-core communication is packetized and multiplexed onto shared wires. As Moore's Law enables billions of transistors on a chip and design complexity and power constraints push for the replication of processor cores on a single chip for both general purpose chip multiprocessors (CMPs) and embedded multiprocessor systems-on-a-chip (MPSoCs), on-chip networks or networks-on-chip (NoCs) are becoming the de-facto communication fabric, with various products in the pipeline [27]. However, no complete traffic model for on-chip networks exists. Designers frequently rely on simple synthetic traffic patterns such as uniform random, bit-permutation and tornado traffic to stress-test a network design [5, 15].

II. TRAFFIC MODEL

Network traffic models have evolved significantly over the lifetime of the Internet. The earliest models were largely Poisson-based, designed for ease of analysis. The discovery of traffic self-similarity and long-range dependence required significant changes to traffic models. We examine this evolution from the early models, through the discovery of self-similarity, to the modern systems that account for complexities such as the impact of TCP congestion control, and practical modeling of arbitrary network traffic.

Network traffic modeling is used as the basis for the design of network applications and for capacity planning of networking systems. Given the impact of poor choices in this arena, it is clear that the validity of the underlying models is of critical importance. Most analytical queuing models of networking systems, such as switches, are driven by a mathematical description of the network traffic.

The factors used to evaluate a system are taken directly from the underlying traffic model. This use requires that traffic models be both valid, resembling reality closely, and sufficiently simple as to allow queuing analysis models to reach a steady-state.

There are two major parameters generated by network traffic models: packet length distributions and packet inter-arrival distributions. Other parameters, such as routes, distribution of destinations, etc., are of less importance. Simulations that use traces generated by network traffic models usually examine a single node in the network, such as a router or switch; factors that depend on specific network topologies or routing information are specific to those topologies and simulations.

The problem of packet inter-arrival distribution is much more difficult. Understanding of network traffic has evolved significantly over the years, leading to a series of evolutions in network traffic models.

III. BACKGROUND

Poisson models have been in use in the literature since the advent of computer networks, and before that in the telecommunications arena [Mandelbrot65]. Memoryless models are very attractive from an analytical point of view [Frost94], and with proper selection of parameters a Poisson model can be fit to most network traffic traces reasonably well for short periods.

Pure Poisson traffic models have a single parameter, the mean arrival rate (λ). Thus, packet inter-arrival times have an exponential distribution with mean $\frac{1}{\lambda}$. Packet arrivals over an interval *[t1, t2]* with *t=t2-t1* have a Poisson distribution with mean $t\lambda$.

Poisson models are popular in queuing theory because they have attractive analytical properties: they are memoryless, meaning that future behavior has no link to past behavior, recent or distant; and aggregation of multiple Poisson streams generates a new Poisson stream with $\lambda'=\Sigma\lambda$. Poisson queuing models have an easily derived steady state equation $p_n = \frac{\lambda^n}{\mu^n} p_0$, where μ is the departure rate and p_0 is the probability of an empty system. This analysis is significantly more complex in other models.

It has been suggested [Frost94] that the networking community has held on to the Poisson model in the face of contrary evidence because of these attractive properties.

In the compound Poisson model, the base Poisson model is extended to deliver *batches* of packets at once. The inter-batch arrival times are exponentially distributed, while the batch size is geometric.

Mathematically, this model has two parameters, λ, the arrival rate, and ρ in *(0,1)*, the batch parameter. Thus, the mean number of packets in a batch is $\frac{1}{\rho}$, while the mean inter-batch arrival time is $\frac{1}{\lambda}$. Mean packet arrivals over time period t are $\frac{t\lambda}{\rho}$.

The compound Poisson model shares some of the analytical benefits of the pure Poisson model: the model is still memoryless, aggregation of streams is still (compound) Poisson, and the steady-state equation is still reasonably simple to calculate, although varying batch parameters for differing flows would complicate the derivation.

One complaint with compound Poisson models is that truly back-to-back packet arrivals are rare in practice [Jain86]. This may no longer be the case. The argument in 1986 was that most network nodes were shared by nonnetwork activities, and that packet processing generally took longer than packet transmission times. Thus, the time between packets from a single node should be fairly large. However, modern CPUs are usually fast enough to saturate the outgoing network link even under non-network load.

Another attempt at providing a bursty traffic model is found in Jain and Routhier's Packet Trains model [Jain86]. This model was principally designed to recognize that address locality applies to routing decisions; that is, packets that arrive near each other in time are frequently going to the same destination. In generating a traffic model that allows for easier analysis of locality, the authors created the notion of *packet trains*, a sequence of packets from the same source, traveling to the same destination (with replies in the opposite direction).

Packet trains are optionally sub-divided into *tandem trailers*. Traffic between a source and a destination usually consists of a series of messages back and forth. Thus, a series of packets go one direction, followed by one or more reply packets, followed by a new series in the initial direction. The authors designate these series of packets as tandem trailers within a single train and analyze the trailer characteristics as well as the overall train characteristics.

Traffic quantity is then a superposition of packet trains, which generates substantial bursty behavior. This refines the general conception of the compound Poisson model, which recognized that packets arrived in groups, by analyzing *why* they arrive in groups, and better characterizing the attributes of the group. Finally, the authors clearly demonstrate that packet arrival times are *not* Poisson distributed, which led to a model that departs from variations on the Poisson theme.

Unfortunately, this makes mathematical analysis of a system difficult using the packet train model. Much more work would be necessary before this model could be used in practical application evaluation. At best, this provides a roadmap and a framework for detailed analysis. For example, a new protocol designer may be unable to determine some of the parameters of the packet train model of the new protocol without a prototype of the protocol.

Finally, the packet train model is extremely detailed — so detailed that analysis of systems may be prohibitively difficult. For example, a designer of a switch wants to know the amount of buffer space necessary to reduce congestive losses to an acceptable level. However, a complete characterization of all traffic that must be supported would be needed before the analysis could be undertaken.

So, now the idea of Bernoulli traffic model has been introduced in this paper. This Bernoulli traffic generator will overcome all disadvantages of the poisson traffic model.

IV. TRAFFIC GENERATOR

A traffic generator system consists of a MASTER TG and SLAVE TGs as shown in Figure 2. The master performs the operation of a TG like the slaves but, it has an extra logic to take reports from the slaves inorder to connect with the HOST PC.

Every TG has an ID or address which is 7 bits long and range from 0 to 127. The master will always take the address 0. Configuration and reporting data is done via the NoC under evaluation. A global extra signal shows whether the system is running or in the configuration /reporting state. Every TG has clock cycle counter synchronized to others, this can be used to calculate delays. Every word is counted both in sender and reader, and normal data words carries the lowest bits of this sequence number. Missed and duplicated words and out-of-order errors are detected.

The inner working of TG is shown in Figure 1. When system is resetted, global configuration mode enable signal is ‘1’, so system resets in configuration state. Add trigger commands are given via UART. Master interprets commands and sends them to appropriate TGs using simpler protocol. When start command is given, master sets configuration mode enable to ‘0’ the system starts running. When running time is over, master sets configuration mode enable to ‘1’, the slaves read every data present for a given time to empty the network from benchmarking data. When the network emptying time has reached, master sends reporting command to first TG. The TG sends the report data to master using simple protocol. Then the master prints report and gives reporting command to next TG, until every report has been asked and printed. If there is a NoC monitor in use, the master gives UART to that monitor and gives reporting command for monitor.

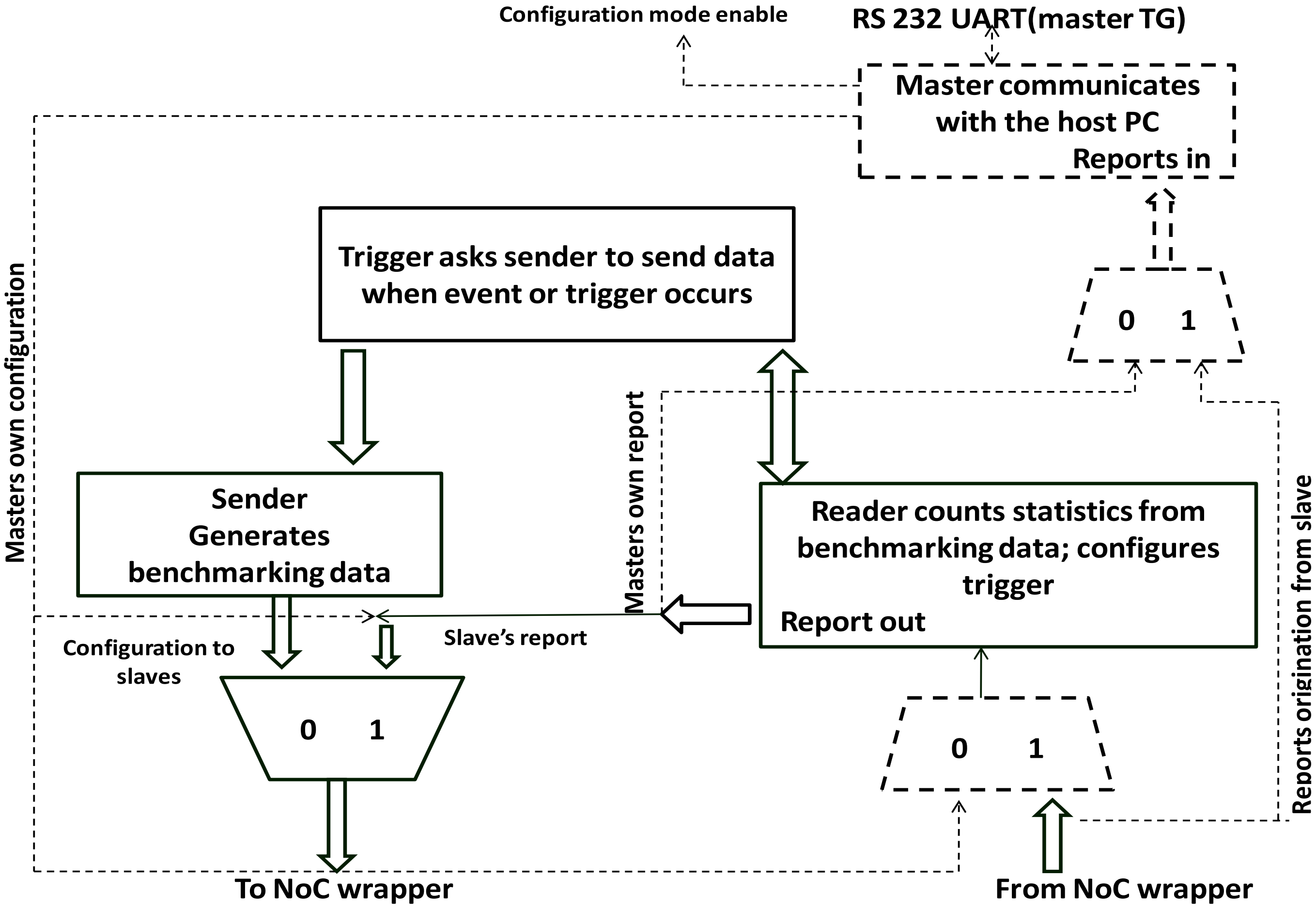

Figure 1: Inner working of Traffic Generator

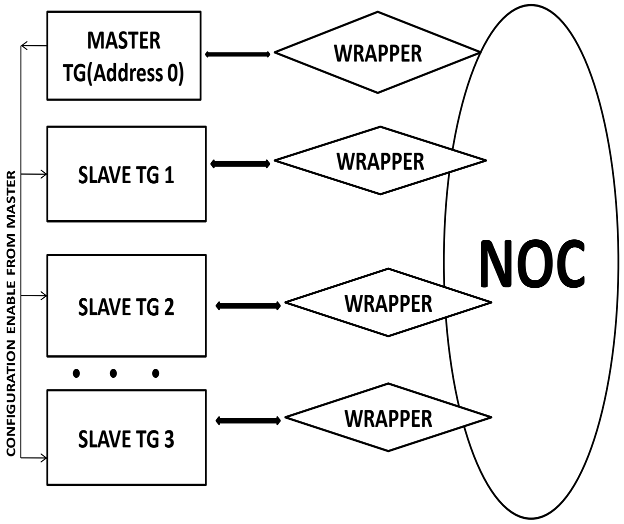

Figure 1: Traffic generator with NoC

V. BERNOULII'S RANDOM NUMBER GENERATION

The Bernoulli Binary Generator block generates random binary numbers using a Bernoulli distribution which is shown in equation 5.1. The Bernoulli distribution with parameter p produces zero with probability p and one with probability 1-p. The Bernoulli distribution has mean value 1-p and variance p(1-p). The Probability of a zero parameter specifies p, and can be any real number between zero and one.

Syntax

x = RAND('BERNOULLI',p)

where

x is an observation from the distribution with the following probability density function:

$$f(x)=\begin{cases}1 & p=0, x=0\\ p^{x}(1-p)^{1-x} & 0<p<1, x=0,1\\ 1 & p=1, x=1\end{cases}$$

Range: $x = 0, 1$

P is a numeric probability of success.

Range: $0 \le p \le 1$

VI. PROPOSED TRAFFIC GENERATOR DESIGN

A traffic generator has been designed in order to generate traffic in the NoC. The traffic generator design is shown below in Figure 3.

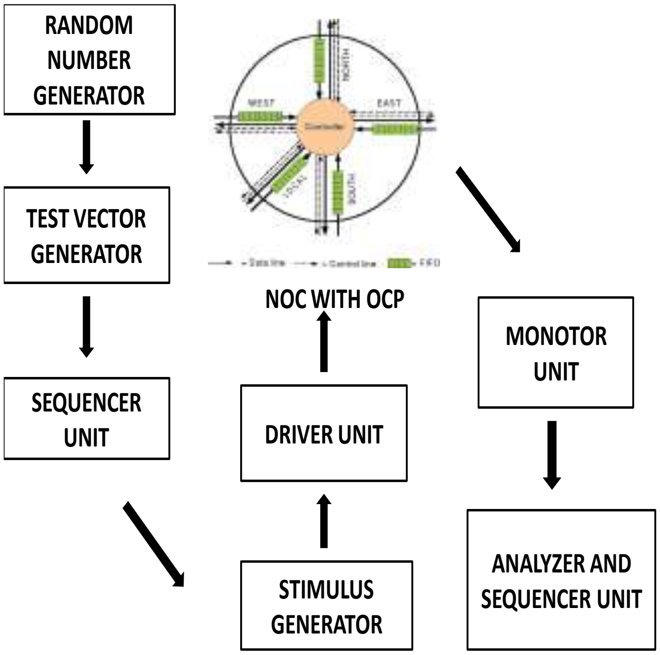

Figure 3: Proposed Bernoulli traffic generator design

The traffic generator consists of a random number generator. The random number is generated by Bernoulli's process. These random numbers are converted into test vectors by the LFSR in the test vector generator. The sequencer unit lines up the test vectors. The test vectors are sent to the router in the sequence given by sequencer. The stimulus generator gives the direction in which data flow should take place. The monitor unit is used to show the data transfer and direction of data flow. The analyzer and coverage collector unit is used to analyze whether the data received is correct. The VHDL coding to realize the design is written. The output of which can be viewed only on hardware implementation using the Altera Quartus tool on Cyclone III board.

VII. CONCLUSION

The Bernoulli traffic generator is capable of automatically selecting the sender and receiver cores from the given set of cores. Then test vectors are generated using an LFSR. These test vectors are considered to be data to be transferred within the NoC. The sequence and direction of data flow are also decided by the Bernoulli traffic generator. This traffic generator when interfaced with an NoC will generate automatic traffic in the NoC , which can be used to check the working of NoC. Also the TG is capable of showing which cores are participating in the transmission currently. Finally the analyser unit in the TG shows any transmission failures.

VIII. FUTURE WORK

The Bernoulli traffic generator can be made to generate traffic in a NoC with router as interconnect. Hardware implementation of the TG and NoC will show better results. The extension of this work will be done shortly and will be given in the next paper.

IX.REFERENCES

[1] A. Balachandran et al. Characterizing user behavior and network performance in a public wireless LAN. In Proc. of the ACM SIGMETRICS international conference on Measurement and modeling of computer systems, pp. 195–205, June 2002.

[2] L. Benini and G. De Micheli. Networks on chips: a new SoC paradigm. In IEEE Computer, Vol. 35, No. 1, pp. 70–78, Jan. 2002.

[3] I. Y. Bucher and D. A. Calahan. Models of access delays in multiprocessor memories. In IEEE Transactions on Parallel and Distributed Systems (TPDS), Vol. 3, No. 3, pp. 270-280, May 1992.

[4] W. J. Dally and B. P. Towles. Principles and practices of interconnection networks. Morgan Kaufmann Publishers, ISBN: 0122007514, San Francisco, CA, 2004.

[5] W. J. Dally and B. Towles. Route packets, not wires: on-chip interconnection networks. In Proc. of the 41st Design Automation Conference (DAC'01), pp. 684–689, June 2001.

[6] F. Darema-Rogers et al. Memory access patterns of parallel scientific programs. ACM SIGMETRICS Performance Evaluation Review, Vol. 15, No. 1, pp. 46–58, May 1997.

[7] P. Doukhan et al. Theory and applications of long-range dependence. Birkh¨auser, ISBN: 0817641688. Switzerland, Dec. 2002.

[8] W. E. Leland et al. On the self-similar nature of Ethernet traffic (extended edition). IEEE/ACM Transactions on Networking, Vol. 2, No. 1, pp. 1–15, Feb. 1994.

[9] K. Lahiri et al. Evaluation of the traffic-performance characteristics of system-on-chip communication architectures. In Proc. of the 14th International Conference on VLSI Design, pp. 29–35, Oct. 2000.

[10] U¨ . Y. Ogras et al. Key research problems in NoC design: a holistic perspective. In Proc. of the 3rd IEEE/ACM/IFIP International Conference on Hardware-Software Codesign and System Synthesis (CODES+ISSS'05), pp. 69–74, Sept. 2005.

[11] K. Park andW.Willinger. Self-similar network traffic and performance evaluation. John Wiley and Sons, ISBN: 0471319740. New York, NY, 2000.

[12] V. Paxson and S. Floyd. Wide-area traffic: the failure of Poisson modeling. IEEE/ACM Transactions on Networking, Vol. 3, No. 3, pp. 226– 244, June 1995.

[13] D. Pham et al. The design and implementation of a first-generation Cell processor. In Proc. of the International Solid State Circuits Conference (ISSCC'05), pp. 184–185, March 2005.

[14] M. Sgroi et al. Addressing the System-on-a-Chip interconnect woes through communication-based design. In Proc. of the 41st Design Automation Conference (DAC'01), pp. 667–672, June, 2001.

[15] S. Turner. Performance analysis of multiprocessor interconnection netwrosk using a burst-traffic model. Ph.D. Dissertation Thesis, University of Illinois at Urbana-Champaign, 1995.