

NoCVision: A Network-on-Chip Dynamic Visualization Solution

Vaibhav Gogte[†], Doowon Lee[†], Ritesh Parikh[‡] and Valeria Bertacco[†]

[†]University of Michigan
Ann Arbor, MI, USA
{vgogte,doowon,valeria}@umich.edu

[‡]Intel Corporation
Santa Clara, CA, USA
ritesh.parikh@intel.com

ABSTRACT

Networks-on-chip (NoCs) are the communication infrastructure of choice for integrating the many components of modern silicon systems, deployed anywhere from systems-on-chip, to chip multi-processors, and to heterogeneous systems. The growing design complexity of these systems, coupled with shrinking times-to-market, requires efficient analysis of complex applications mapped onto the network in a short span of time. In this work, we propose NoCVision, a novel platform for the analysis of NoC characteristics and traffic flows. NoCVision enables design-space exploration, performance tuning, and validation of the NoC subsystem. It allows to consolidate and summarize the network's simulation data and visualize it through intuitive diagrams and plots, either in a static form or animating it to depict changes occurring over time during an application's execution. To showcase the features and benefits of NoCVision, we present several case studies developed on a 64-node CMP organized in a 8x8 mesh NoC and running multi-programmed workloads.

Categories and Subject Descriptors

C.1.2 [PROCESSOR ARCHITECTURES]: Multiprocessors—*Interconnection Architectures*; I.6 [SIMULATION AND MODELING]: Simulation Output Analysis

General Terms

Performance, Verification, Measurement

Keywords

visualization, design analysis, verification, design-space exploration

1. INTRODUCTION

Networks-on-chips (NoCs) are increasingly being deployed in modern systems-on-chip (SoCs) and chip-multiprocessors (CMPs) for integrating and facilitating communication between cores, memories, and application-specific components. The complexity of these NoC designs is also on the rise, to cater to the communication requirements of these highly parallel architectures. During the development of these systems, extensive effort is dedicated to running simulations

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

NoCArc '15, December 05 2015, Waikiki, HI, USA
©2015 ACM. ISBN 978-1-4503-3963-6/15/12...\$15.00
DOI: <http://dx.doi.org/10.1145/2835512.2835518>

for the purpose of design exploration and evaluation of the performance and verification of the NoC design. However, the vast complexity of these systems, with hundreds of heterogeneous components connected to the interconnect, leads to an impractical number of distinct execution scenarios to be evaluated. As a result, simulations span billions of clock cycles generating a vast amount of data to be analyzed. To continue to support the growing design complexity of these systems, in the face of shrinking time-to-market, novel solutions are required to enable a faster, more flexible, and intuitive approach to analyze and consolidate the data collected during simulation, verification or post-silicon tuning.

In this work, we present NoCVision - a tool that aids in efficiently analyzing NoC traffic activity. The tool synthesizes the traffic data gathered in simulation, whether with an architectural, a RTL or a post-silicon simulator, and presents it through intuitive diagrams and plots that allow designers to quickly understand and analyze the behavior and activity of the NoC under evaluation. With reference to Figure 1, NoCVision takes in input log data gathered in simulation, parses it to extract the relevant information, and visualizes user-selected *metrics* in its rich graphic user interface. Data can be plotted and charted using a range of formats. It can be coalesced to coarse time window granularities, or in fine-grained time steps. NoCVision is capable of visualizing any metric defined by the user, and that can be associated with either links or routers or virtual channels in the network. Moreover, it also employs a rich set of interactive controls enabling intuitive and flexible navigation through the data under investigation. It thus provides a platform for design-space exploration, optimization, verification, and performance and power tuning of NoC designs. NoCVision is publicly available online for download.

The novel **contributions** of our solution are:

- A novel network traffic analysis and visualization solution that allows to analyze and present traffic metrics through intuitive and synthetic charts and plots. NoCVision analyzes data from network model's simulations and presents it to the user through a rich and flexible visual interface.
- Our tool is applicable to a wide variety of NoC simulation frameworks, as it can be set up to report on any user-set network metric that can be mapped to links, routers or virtual channels.
- NoCVision allows a user to move back and forth in time in visualizing the relevant network metrics, at any level of granularity, down to the single clock cycle. This feature enables users to animate sequences of snapshots and appreciate activity over time.

2. RELATED WORK

Many previous research efforts have focused on simulating the network under different traffic scenarios. NoC simulators such as BookSim [7], SICOSYS [10], Noxim [2] and

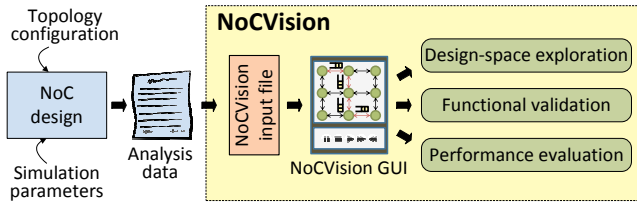


Figure 1: **NoCVision analysis flow.** Data generated from NoC simulations or design frameworks is processed by NoCVision, which condenses it, extracting relevant metrics, and presents it through intuitive diagrams and plots. NoCVision enables NoC’s design-space exploration, functional validation and performance evaluation.

GARNET [1] provide flexible network infrastructures and are widely used by researchers today in exploring and analyzing various network components. While these simulators provide a framework for logging data, they do not provide a way to analyze the data efficiently. They are often capable of recording the traffic information in the form of output logs but they do not support the compilation of data. Moreover, it is impractical to extract relevant information from the data collected by these simulators due to the extreme size of the data logs.

In addition, there exists only a few solutions that specifically aid in analyzing NoC designs. NOCEVE [5] is one such tool that monitors NoC traffic on FPGA platforms. It can be used for the analysis and verification of applications running FPGA prototypes of SoC designs. However, it cannot operate on software simulations of such designs and hence cannot be used for the analysis of traffic flow occurring in NoC simulators. MAIA [8] is yet another simulator that generates an NoC model from parameterizable templates, simulates the traffic, and analyzes the reports generated from the simulations. MAIA’s generated traffic logs provide traffic information, such as number of packets received and average packet delivery time, which are valuable for design analysis. On the other hand, NoCVision enables a better understanding of such reports by consolidating and displaying the collected data graphically. Similar work has also been pursued for off-chip networks and Nam [4] is one such tool, used for animating packet movements in physical network traces. However, to the best of our knowledge, NoCVision is the first stand-alone solution that aids in the analysis of on-chip traffic by visualizing traces and network metrics.

3. OVERVIEW

NoCVision is a software tool to aid in the analysis of NoC designs. It takes in input data logs generated by design simulations and provides interactive controls to analyze the behavior of NoC designs. NoCVision is oblivious to the semantics of the network metric that is under analysis, thus it can chart any network metric that can be associated with individual routers, links or virtual channels. Some of the network properties that can be analyzed with NoCVision include congestion, average component utilization, power state, fault status, and individual packet movements through the network.

NoCVision visualizes these metrics through diagrams and plots and is capable of tracking their status and values throughout the duration of a simulation. Each individual diagram presents the network metric using varying color intensities, similar to a heat map. If the analysis targets binary metrics, such as faulty routers or packet movements, then the corresponding diagram is simply bicolor. For instance, while plot-

ting average utilization of routers, heavily utilized routers are represented with brighter colors, in contrast with those that are sparsely utilized, which are shown in light color. Time is managed in *intervals* of fixed length. When appropriate, the metric under study is accrued over the entire interval before it is displayed. The size of each interval is set by the user and it can be varied to span anywhere from the entire simulation, down to the individual clock cycle. NoCVision also provides the capability of displaying specific interesting incidents, which occurred during the simulation, called *events*.

Figure 1 illustrates the network analysis flow with NoCVision. The network design under evaluation is simulated with synthetic workloads, verification testcases, or commercial benchmark suites. Data is then recorded either directly from the simulation flow, or through log-generation scripts, and transformed into a *NoCVision input file*, which contains all the information that NoCVision uses to operate. The NoCVision visualizer is then invoked to graphically analyze the relevant metrics and events captured in the log.

4. COMPONENTS OF NOCVISION

NoCVision considers the data gathered in simulations and classifies the properties to be analyzed as network metrics. These metrics are user-defined and are used to identify the type of data to be extracted or visualized. Depending on the type of analysis to be performed, NoCVision can operate in one of three modes: interval, plot, or event mode, as discussed in more detail in Section 5. The *plot mode* aids in simultaneously plotting multiple metrics over time on a same graph. Moreover, it can show the dynamic difference between metrics to facilitate a comparative analysis. In contrast, *interval mode* plots one or a few metrics, one snapshot per interval, using heat maps matching the value of the metric for each network component (routers, links or virtual channels). In interval mode, NoCVision highlights anomalous regions in the network and allows their inspection at varying time interval granularities. Once the region and period of interest are determined, *event mode* can expose the activity that triggers the anomaly allowing a fine grain inspection down to the single clock cycle.

NoCVision is implemented in Python and uses its built-in GUI library *tkinter* to graphically display the animation of traffic flow. Functionalities in the NoCVision GUI are implemented using the standard *tkinter* library.

4.1 NoCVision input file

The NoCVision input file specifies the topology and configuration of the physical network and the data to be visualized. The syntax of the input file is shown in Figure 2 and an example is presented in Figure 3. The network configuration is specified by topology, dimension k , number of nodes in each dimension n and number of virtual channels num_vcs . For instance, the example of Figure 3 specifies a 4x4 2D mesh topology, supporting 16 virtual channels per port. Note that currently NoCVision supports only regular topologies, such as 2-dimensional mesh and torus.

Following the topology configuration, the input file includes the data relative to the network metrics to be analyzed. The file contains a series of sections, each section titled with the time at the end of the interval for which the file reports data, followed by the values of all the metrics collected for that interval (Figure 2 (line 4-6)). For instance, the example of Figure 3 (line 5-8) reports two values of global average network latency *avg_latency*, one for

Figure 2 NoCVision input file syntax

1: NoCVisionConfig → NetworkConfig DataLog ⁺
2: NetworkConfig → topology k n num_vcs
3: DataLog → IntervalLog EventLog
4: IntervalLog → IntervalHeader IntervalMetric ⁺
5: IntervalHeader → [clock = integer]
6: IntervalMetric → metricName = integer float array
7: EventLog → clock_integer:id_integer:Event Type: EventLocation
8: EventType → trace pin unpin
9: EventLocation → node_integer node_integer: link_integer node_integer: link_integer: vc_integer

the interval 1-100 clock cycles, and one for the interval 101-200. Metrics associated with different network components can be logged simultaneously within these sections. NoCVision infers whether the metric is global or associated with routers, links or virtual channels based on the size of the array reporting the metric’s values. Table 1 reports the format of each type of metric. For instance, in a 3x3 mesh network, metrics associated with links would be reported in bi-dimensional arrays of size 4x9, as there are nine routers in the network and each router has a link in each of the four directions. It is also possible to take into account local links in reporting link-metrics and provide arrays of values accordingly. In this latter case, the value associated with the local link will be displayed on the corresponding router.

Reference network component	Metric input syntax
Network	value
Router	[array of #routers values]
Link	[[array of #links/router] * #routers values]
Virtual Channel	[[[array of #VCs/port] * #links/router] * #routers values]

Table 1: **Format of metrics’ values reported in the NoCVision input file for the interval mode**

Following the metrics associated with intervals, which are visualized using the plot and interval modes, the NoCVision input file reports the event data, relevant for event mode. This section consists of a list of events, where each event is described by the time at which it occurred, an event ID, the type of event and its location, in terms of router, link and VC. Currently, we support three types of events: trace, pin and unpin. The example of Figure 3 shows a trace event occurring at cycle 143, with event ID 3, in router 2, on its north link (link 0). The semantic of each type of event is discussed in Section 5.3. Note that the input file can include both events and interval-related data, however, the information shown at any point in time depends on the visualization mode selected dynamically for NoCVision.

While the current version of NoCVision can analyze the activity of routers, links and virtual channels, in future versions, we plan to support finer-granularity elements such as pipeline stages within a router.

We have integrated a cycle-accurate network-on-chip simulator [7] with NoCVision.¹ The integration requires some simple instrumentations in the simulator source code, so that the desired simulation data can be recorded in the analysis

¹Instructions on how to integrate NoCVision are available at: <http://nocvision.eecs.umich.edu/>.

data file. Note that, to reduce logging overheads, data can be collected for the regions of interest only.

Figure 3 Example of NoCVision input file

1: topology = mesh	
2: k = 2	
3: n = 4	
4: num_vcs = 16	▷ Refer to figure 2: line 1-2
5: [clock = 100]	
6: avg_latency = 43.3	
7: [clock = 200]	
8: avg_latency = 56.2	▷ Refer to figure 2: line 4-6
9: clock_143 : id_3 : trace :	
node_2 : link_0	▷ Refer to figure 2: line 7-9

4.2 Graphical user interface

The NoCVision GUI provides a number of capabilities, depending on the mode of operation. When in interval mode, it is possible to display the specific metric’s values associated with routers and links corresponding to each interval, information related to the virtual channels and a color bar showing the range of values a metric assumed throughout the interval. In plot mode, it is possible to plot the values of a metric associated to a network component over the entire simulation, and it is possible to compare multiple metrics by plotting them in the same chart through a difference plot. Finally, for both event and interval modes, it is possible to skip, combine or fast-forward through events or intervals.

When NoCVision starts, it parses through the input file to identify all the modes for which data is provided, which it then presents to the user. The interface is customized for each mode selected, to enable the features pertinent to that mode. The user can then select the specific options of interest and visualize the data. Figure 4 shows a sample layout of the GUI when NoCVision is in the interval mode. The user can choose the metric to be plotted, the time window size, etc., and then view the data on the network diagram. Data is visualized through color maps; in addition, the diagram is interactive and routers, links and VC can be selected to visualize the data values or to generate a plot of the metric’s values associated with the component. The user can also control the playback rate in traversing the simulation through the control buttons provided. Finally, controls are provided to restart the analysis or present a new set of metrics and options to the user.

5. MODES OF OPERATION

NoCVision operates in one of three modes, depending on the type of data under analysis. Each mode can operate at a range of time granularities, thus allowing a user to first gather a summary view of the activity, and successively dissect it at finer and finer time scales.

5.1 Interval mode

The interval mode presents the target metrics summarized through the time interval selected. The NoCVision’s visualizer allows users to coalesce data over multiple intervals and to begin the analysis from any interval, possibly one deep in the simulation. When coalescing metrics’ data, several aggregation options are available, namely (i) ‘accumulate’, providing the sum over all the time intervals to be grouped together, (ii) ‘maximum’, or (iii) ‘minimum’, using the largest or smallest value observed, and (iv) ‘average’, which computes the average of the values. These later aggregators are particularly useful when plotting network metrics such as average injection rate or maximum packet latency.

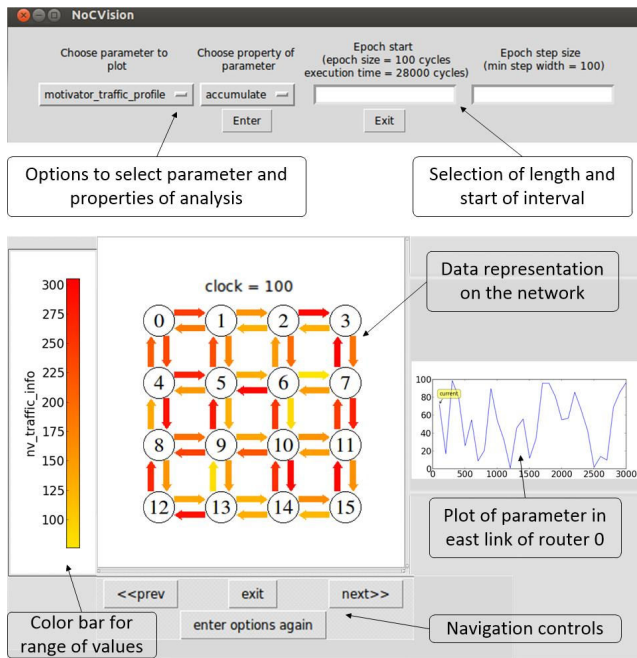


Figure 4: **NoCVision interval-mode layout.** After a user selects which metric to analyze and the time interval to use, NoCVision provides an interactive diagram for the metric and allows detailed plots and backward and forward animations.

After the user selects a network metric and an aggregator, NoCVision displays the metric using color intensity to visualize relative values in the network diagram, and it provides a color bar as a legend to associate values to the colors. With reference to the example of Figure 4, the color intensity varies from *yellow* for smaller values to *red* for larger values of the target metric. The diagrams in NoCVision are interactive, so it is possible to select any component to visualize the value of the metric associated with it during the present interval or generate a graph of the value over the entire simulation. Finally, the NoCVision interface allows the user to move forward and backward in time and fast-forward through time intervals to quickly reach regions of interest.

5.2 Plot mode

In plot mode, NoCVision can plot several metrics simultaneously on the same chart and it also allows plotting the difference between a pair of metrics. For instance, a user may be interested in plotting the average network throughput (a network-level metric) over the entire simulation execution, or in inspecting metrics associated with routers or links. NoCVision can plot up to three types metrics simultaneously, one associated with links, one with routers and a network-wide one. Figure 5 shows, in the top part, an example of metrics selection in plot mode: a user is presented with the list of metrics available – gathered from the NoCVision input file – and can select the three of interest. Note that once a metric is selected, it is possible to choose to plot the values associated with any number of components of that type. For instance, with reference to Figure 5, the user selected to plot the metric *nv_router_inflow* for both routers 0 and 4, along with other link plots. The plot then uses color to differentiate among the selected metrics.

Some of the features of plot mode include the ability to (i) zoom the chart along both the value and time axes, (ii) inspecting each value point in the plot, (iii) saving the plot in a standard format, such as jpeg or eps. Moreover, plot

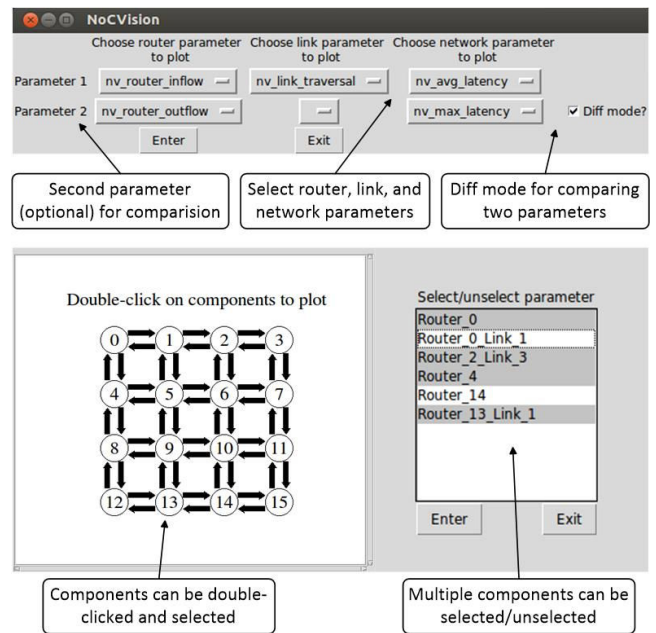


Figure 5: **Layout of NoCVision's plot mode.** In plot mode, users must select the metrics they wish to chart (top part), and whenever they require a diff plot with other metrics. NoCVision then presents an interactive diagram where network components can be selected so that their associated values – for the metric selected – are plotted over the entire span of the simulation.

mode allows also to chart the difference between two metrics. This is also illustrated in the top part of Figure 5. This *diff* mode can be used even to compare metrics from different simulation runs. This latter feature becomes often valuable to compare multiple architectural options.

5.3 Event mode

The event mode aids in the visualization of discrete events occurring in the simulation; an event being a binary incident, associated with a router, link, or virtual channel having on-off phases. An event is termed *on* when a metric under analysis is active during the simulation. The event is later disabled once the metric is de-activated. Multiple such events manifesting in the various network components can be recorded in the configuration file. Each event logged must include the clock cycle at which the event was triggered and the location in the network where it manifested. Note that events can be active for any number of clock cycles and be irregularly dispersed in the simulation. NoCVision maps these events onto the network diagram and can display the status of the network at the occurrence of each event. Multiple events can be selected so that network status can be visualized after the triggering of each event. Each event is identified by its unique ID; the NoCVision GUI uses colors to distinguish between the events while mapping them on the network diagram. Depending on how long an event is active, NoCVision deems it an instantaneous or a window event. An instantaneous event is active only for one clock cycle, and it is specified as a *trace* event. Window events, on the other hand, last longer and they must be explicitly activated (*pin*) and deactivated (*unpin*).

Figure 6 shows an application of NoCVision's event mode to evaluate router congestion. In this example, routers use a simple heuristics, based on buffer occupancy, to determine whether they are congested. Event 1 marks when routers 9 to 14 simultaneously become congested. Event 2 observes

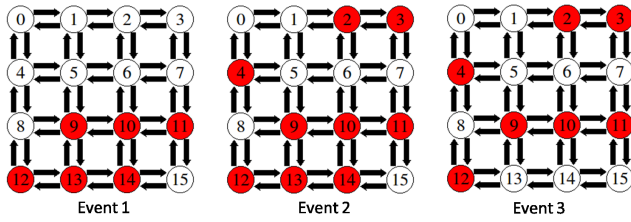


Figure 6: **Example of events in NoCVision.** The diagrams show the onset and dimming of congestion (red) for the routers in the network. Congestion is computed based on buffer occupancy in the routers, and NoCVision’s events are pinned and unpinned accordingly.

three more routers reaching the congested status. Finally, event 3 marks when router 13 and 14 are no longer congested. Note that this example uses *pin* and *unpin* type events to mark the onset and the dimming of congestion, respectively.

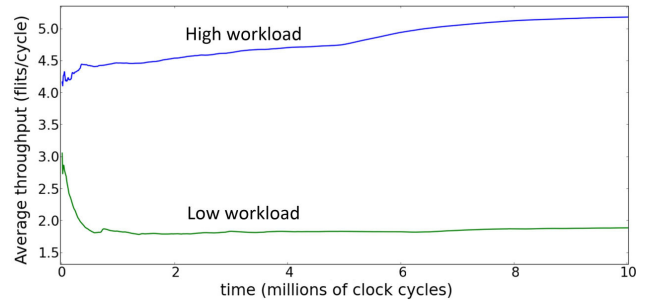
6. EVALUATION

We evaluated NoCVision by integrating it with a cycle-accurate multi-core simulator. We configured an 8x8 mesh NoC with 3-stage pipeline routers, 4 VCs/port and XY routing. We ran the SPEC CPU 2006 benchmark suite and, during each simulation, we logged network traffic information with a range of time granularities. We then evaluated NoCVision by performing the analysis of the network load imposed by a range of workloads derived from mixing SPEC applications. Subsequently, we evaluated the benefits of two recent solutions using NoCVision: an application-aware topology reconfiguration solution, HiROIC [6] and a power-efficiency technique, Panthre [9].

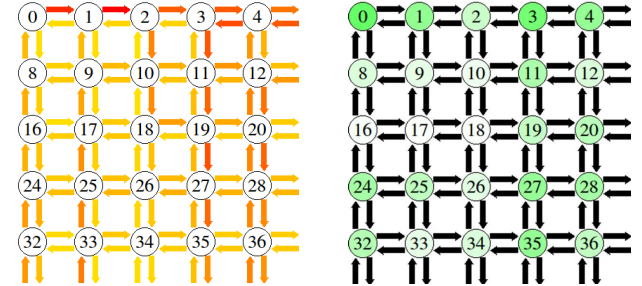
6.1 Application workloads analyses

To analyze this application with NoCVision, we first evaluated the characteristics of our benchmarks when executing on a 64-node chip multi-processor, with 60 processor cores and 4 memory nodes in the corners of the mesh topology. We classified 35 SPEC applications based on their level-1 cache (L1) misses per kilo instructions (MPKI) (as shown in [3]) as this metric directly correlates to generated network traffic. We then generated three workloads, each running a mix of SPEC applications: (i) a workload with high-MPKI applications (*high*), (ii) a workload running low-MPKI applications (*low*), and (iii) a workload running a mix of high-MPKI and low-MPKI applications (*medium*). Figure 7a plots the average throughput measured every 10,000 clock cycles for the high and low workloads. As expected the plot indicates that the high category workloads have a higher throughput.

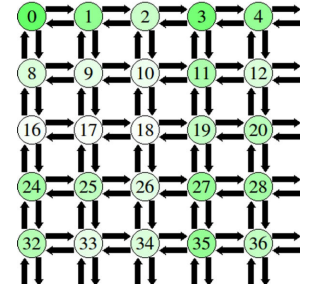
Figures 7b and Figure 7c focus on a medium category workload, where high MPKI applications have been mapped to the central quadrant nodes (27-28, 35-36) and along the edges (3-4, 24, 31-32, 39, 59-60) and low MPKI applications are mapped to the other CMP’s nodes. Figure 7b reports the average link utilization using NoCVision in interval mode, for the entire simulation duration, by using a heat map from yellow to red on the links. Figure 7c reports a router-mapped metric, number of packets injected into the network, using varying green intensity for the routers. Note how the corner nodes, connected to memory controllers, are injecting heavy traffic. Similarly, the nodes executing high-MPKI applications are also producing more network traffic than the others. For reasons of space, we only show a portion of the network diagrams.



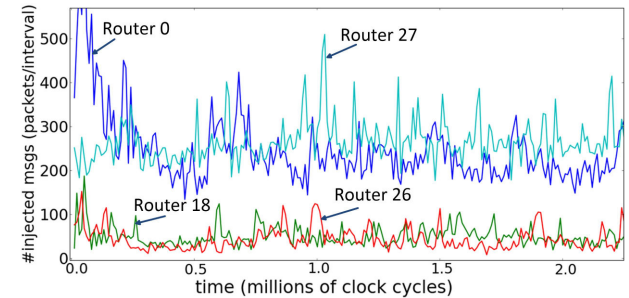
(a) Average throughput for *high* and *low* workloads



(b) Link utilization for *medium* workload (network portion)



(c) Router injection for *medium* workload (network portion)



(d) Number of injected packets over time

Figure 7: **Analysis of SPEC CPU 2006 applications.** NoCVision is used to plot various network characteristics logged from SPEC CPU 2006 applications running on a 2D mesh. The plot mode is used to compare the number of packets injected by different nodes in the network. Moreover, the interval mode displays a relative variation of metrics in network components.

By using plot mode, it is possible to visualize the traffic injected over time for each individual router. To this end, Figure 7d charts this metric for routers 0, 18, 26 and 27. Note how the nodes corresponding to memory controllers (node 0) and high MPKI applications (node 27) inject distinctively more traffic into the network, compared to the other nodes.

6.2 Application-aware routing

In this experiment, we used NoCVision to evaluate the HiROIC solution [6], an application-aware topology reconfiguration technique that dynamically adapts the topology to the application characteristics, striving to provide shorter paths between highly interacting network nodes. The solution is equipped with additional links connecting the nodes, so that it is possible to dynamically create shortcuts between some of the nodes in the network. HiROIC studies the traffic pattern over a time interval and attempts to min-

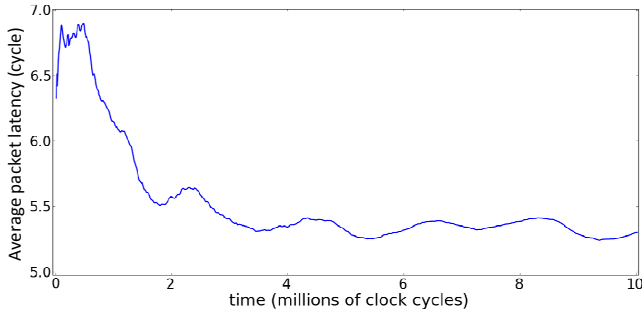


Figure 8: **Analysis of HiROIC solution.** NoCVision plots the difference between the average latency of 2D mesh and HiROIC topologies logged by running the *medium* workload.

imize overall network latency by enabling a selected set of links leading to the smallest hop count between high-traffic nodes. In our evaluation, we compared the average packet latency between a standard 2D mesh topology and a HiROIC topology, using plot mode. Note that the topology of the HiROIC is not a simple mesh or torus, however, we could still apply NoCVision for the analysis in plot mode, since it has the ability to compare traces from distinct simulations. Figure 8 plots the difference in average packet latency between the 2D mesh and the HiROIC topology when running a *medium* workload described above. From the graph, it can be noted that HiROIC provides distinctively better latency. Moreover, note that in the early part of the simulation, this difference is more marked because L1 cache misses lead to higher memory requests, thus throttling the network. HiROIC performs better in this region of simulation as it can reconfigure the topology to facilitate low latency memory accesses.

6.3 Power efficient network analysis

This last study evaluates Panthre[9], a power-efficient solution for NoC designs by leveraging the event mode of NoCVision. The power efficient solution calculates the number of packets traversing routers links over intervals of 10,000 clock cycles (called computational epochs) and determines which are the sparsely utilized regions of the network. It then reconfigures the routing function to power-off infrequently utilized links. Each time the traffic profile changes, Panthre’s routing algorithm computes dynamically a new routing function, by selecting the links to activate. In this evaluation, we used NoCVision in interval mode to analyze the traffic patterns during each execution epoch. In addition, we logged events after each change in the active-links set. Figure 9 reports two example intervals, at 29,000 and 39,000 clock cycles, and the corresponding event at the end of the execution epoch. Note from the figure how the disabled links (in red on the right) correspond indeed to the least loaded links in the interval mode (in yellow on the left). Thus, with NoCVision, we could validate the Panthre’s heuristic in determining which links are the best candidates for energy savings. Note how we could use distinct NoCVision modes together to perform our analysis.

7. CONCLUSION

In this work we presented NoCVision, a visualization tool for networks-on-chip that enables intuitive analyses of a wide range of networks’ aspects, from functional validation, to design-space exploration, to performance optimization, *etc.* The three modes of NoCVision can be used together to analyze NoC designs at a wide range of time granularity.

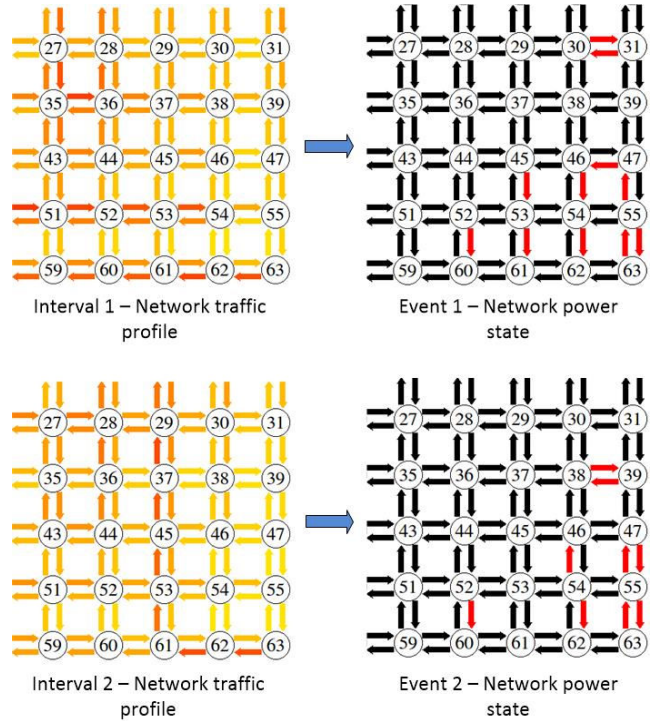


Figure 9: **Link power-off analysis.** NoCVision plots link utilization using the interval mode along with events specifying powered-off links using event mode. Only part of the network is shown for clarity. Powered-off links are shown in *red* while link utilization in interval mode is shown with color intensity varying from *yellow* to *textitred*.

NoCVision is publicly available for open source distribution on the web.

Acknowledgements. This work was supported by C-FAR, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

8. REFERENCES

- [1] N. Agarwal, T. Krishna, L.-S. Peh, and N. Jha, “GARNET: a detailed on-chip network model inside a full-system simulator,” in *Proc. ISPASS*, 2009.
- [2] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, “Noxim: an open, extensible and cycle-accurate network on chip simulator,” in *Proc. ASAP*, 2015.
- [3] R. Das, O. Mutlu, T. Moscibroda, and C. Das, “Aergia: Exploiting packet latency slack in on-chip networks,” *SIGARCH Comput. Archit. News*, vol. 38, no. 3, 2010.
- [4] D. Estrin, M. Handley, J. Heidemann, S. McCanne, Y. Xu, and H. Yu, “Network visualization with Nam, the VINT network animator,” *IEEE Computer*, vol. 33, no. 11, 2000.
- [5] O. Hammami, X. Li, and J.-M. Brault, “NOCEVE: network on chip emulation and verification environment,” in *Proc. DATE*, 2012.
- [6] A. Jain, R. Parikh, and V. Bertacco, “High-radix on-chip networks with low-radix routers,” in *Proc. ICCAD*, 2014.
- [7] N. Jiang, D. Becker, G. Michelogiannakis, J. Balfour, B. Towles, D. Shaw, J. Kim, and W. Dally, “A detailed and flexible cycle-accurate network-on-chip simulator,” in *Proc. ISPASS*, 2013.
- [8] L. Ost, A. Mello, J. Palma, F. Moraes, and N. Calazans, “MAIA: a framework for networks on chip generation and verification,” in *Proc. ASPDAC*, 2005.
- [9] R. Parikh, R. Das, and V. Bertacco, “Power-aware NoCs through routing and topology reconfiguration,” in *Proc. DAC*, 2014.
- [10] V. Puente, J. Gregorio, and R. Beivide, “SICOSYS: an integrated framework for studying interconnection network performance in multiprocessor systems,” in *Proc. EMPDP*, 2002.