Resilient and Efficient Communication in Many-Core Systems using Network Coding

Sadia Moriam*†, Yexin Yan*, Erik Fischer*, Elke Franz‡ and Gerhard P. Fettweis*†
*Dep. of Electrical Engineering and Information Technology / Vodafone Chair Mobile Communication Systems
†Centre for Advancing Electronics Dresden (CFAED)
‡Dep. of Computer Science / Chair of Privacy and Data Security
Technische Universität Dresden, 01062 Dresden, Germany
Email: {sadia.moriam, yexin.yan, elke.franz, gerhard.fettweis}@tu-dresden.de, erik.fischer@vision-in-automation.de

***Abstract*—Due to technology scaling, the number of processor cores on a chip constantly increases. Already today, we reach the domain of so called many-core systems-on-chip. However, this advance comes at the cost of reliability, which especially affects the communication performance of the underlying network-on-chip. Today's resiliency concepts for network-on-chip like automatic repeat request with retransmission are not feasible anymore and lead to long latencies and high network load. In this paper, we propose an on-chip transmission concept based on random linear network coding to provide high resiliency and an efficient communication in many-core processors at the same time. The concept offers a flexible and efficient computable coding scheme, which is well suited for on-chip communication and allows to exploit the path diversity of large networks. First, we use a flit-level cycle-accurate simulation model to investigate the performance potential of the proposed transmission scheme on a network-on-chip with 64 cores. Second, we propose an analytic model for random linear network coding in network-on-chip with retransmission, which is able to provide a very accurate performance estimation close to the cycle-accurate simulation. Finally, we apply the analytic model to investigate the performance potential on the large-scale, assuming a processor with 1024 cores.**

I. Introduction

A. Network-on-Chip

Network-on-chip (NoC) is based on the idea to have a packet switched communication between different modules in a Multi-Processor System-on-Chip (MPSoC) provided by a network of routers [1]. Thereby, a module has to be seen as an abstract definition that can represent a processor core, memory, hardware accelerator or even an external interface. The smallest transmission unit within a NoC is called a flow control digit (flit). Due to its modularity as well as efficient and fast transmission, NoC has been established as interconnection network for modern MPSoC during the last decade [2] [3].

One of the biggest challenges for NoC comes with technology scaling. As long as Moore's Law continues the integration density grows, which allows to put more and more modules on a chip. It is expected that the era of MPSoC turns into many-core with thousands of cores on a chip [4]. Consequently, the interconnection problem becomes challenging. On the one hand, long path latencies as well as low bandwidth per module deteriorate communication performance. On the other hand, technology scaling also leads to decreased communication reliability. Due to dynamic variations of voltage and temperature ”frequent and intermittent soft-errors” are expected [4]. Assuming a large NoC with high flit loss probability, state-of-the-art resiliency mechanisms like Automatic Repeat Request (ARQ) are not sufficient anymore. Long path latencies along with a large number of retransmissions lead to a big drop in NoC communication performance.

B. Network Coding

A promising solution for this dilemma might be provided by network coding [5]. Network coding allows to increase throughput, energy efficiency, and robustness of data transmission. These benefits are implied by the key concept of network coding to compute and send linear combinations of data packets.

In case of Random Linear Network Coding (RLNC), the coefficients for the linear combinations are selected at random [6]. Chou et al. introduced an implementation of this approach that allows for a decentralized solution [7]. The data packets $\boldsymbol{p}_i$ are organized in matrices (generations) of size G. Further, the data packets are amended by a global encoding vector $\boldsymbol{\beta}_i = (\beta_{i,1}, \beta_{i,2}, \ldots, \beta_{i,G})$ with $\beta_{i,j=i} = 1$ and $\beta_{i,j\neq i} = 0$. Linear combinations are only computed for packets of one generation (intra-session network coding). Since the operations are applied to all symbols of a packet, the coefficients of the global encoding vector reflect all operations. Hence, the receiver can decode the data without an explicit transmission of the coding coefficients. Decoding is done by solving the system of linear equations established by the matrix of received data packets. If the sender has sent $C > G$ linear combinations, decoding is still possible in case of packet loss as long as the receiver got any G linear independent data packets.

This work is partly supported by the German Research Foundation (DFG) in the CRC 912 “Highly Adaptive Energy-Efficient Computing” and within the Cluster of Excellence “Center for Advancing Electronics Dresden”.

978-1-4673-8590-9/15/$31.00 ©2015 IEEE

In view of the challenging interconnection problem for future many-core systems, the use of network coding for communication within NoCs seems to be promising. In recent years, some authors have already studied the applicability of network coding for NoCs. Indrusiak investigated the feasibility of network coding by means of the well-known butterfly-network for a multicast scenario with two receivers [8]. He suggested algorithms for finding butterfly arrangements on a 2D mesh and a heuristic for the evaluation of the established butterfly. Further algorithms for the mapping of a butterfly network to a 2D mesh were suggested in [9]. Duong-Ba et al. introduced a possible implementation of a network-coded NoC [10]. They suggested a router architecture and an appropriate flit structure. Vonbun et al. investigated theoretical bounds in terms of hop count improvements of network coded NoCs in comparison to classical dimension-routing NoCs [11].

Overall, these studies indicate that network coding can improve the efficiency of communication within NoCs. However, the existing results consider 2D mesh networks of rather small dimensions (up to 12×12 mesh). Further, they focus mainly on multicast communication based on the butterfly structure.

Within this paper, we focus on investigating the benefits of network coding for unicast transmissions. As network coding scheme, we assume RLNC implemented according to [7]. Figure 1 illustrates the principle of a network coded unicast transmission. Instead of sending an initial transmission, waiting for an ARQ and sending a retransmission, additional redundant flits computed as linear combinations from the current generation are directly sent to overcome flit loss (up to a certain level). This approach has some advantages.

1) In case of a flit loss, we can expect less transmissions as in the uncoded case. This case is demonstrated by an example in Fig. 1. Instead of three transmissions, the path between the two communicating modules has to be traveled only once. Hence, latency and network load is decreased significantly.
2) The coded flits can be sent using alternative paths to exploit spatial diversity and further increase reliability.

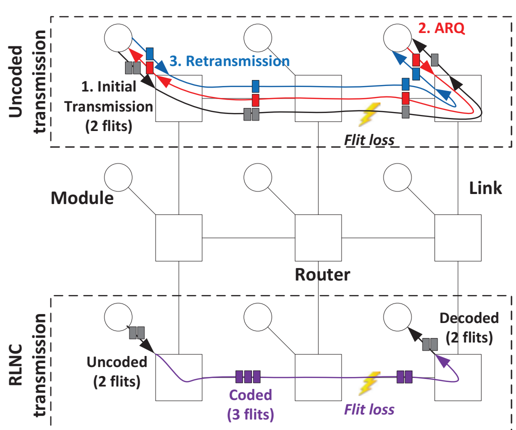


Fig. 1. NoC with RLNC setup vs. ARQ and retransmission.

3) The RLNC code rate can flexibly and dynamically be adapted according to the conditions and requirements of the transmission.

We evaluate the network coded transmission by means of simulations and an analytical model. The analytical model allows to investigate the performance gains for a large-scale network with 1024 cores.

The paper is organized as follows. In Section II, we introduce the system model. The simulation setup is described in Section III. Afterwards, we present results of the simulation in Section IV. Section V introduces the analytical model and its results. Finally, Section VI concludes and gives an outlook on future work.

II. System Model

The following system model is assumed for the subsequent numerical simulations and analytic model.

Topology: The investigation has been limited to the widespread 2D mesh topology for many-core NoC with up to 1024 modules.

Traffic: A spatial uniform traffic distribution with a constant injection probability per module (i.e. Poisson arrival) is assumed. This generic traffic pattern is commonly used for NoC performance evaluation and enables a good comparison with other publications.

Routing: A dimension-ordered, deterministic XY routing is applied. Oblivious routing schemes, like ”Valiant’s random routing” [12], promise for higher resiliency due to the gained spatial diversity, especially in combination with RLNC. However, previous investigations show that this advantage does not take any effect under the uniform traffic assumption. Moreover, the oblivious routing schemes even decrease the achievable throughput due to higher traffic load at the center routers. Finally, the disadvantages of oblivious routing schemes predominate for which reason they will not be considered here.

Coding: Uncoded and RLNC coded transmissions are considered and compared in the following. In case of RLNC, a certain number G (generation size) of subsequent flits is composed as a frame (generation). Out of these G flits, C linear combinations are generated at the sender. Only intra-session network coding is allowed. By applying a sufficient symbol size, it can be assumed that all combinations are linear independent. Thus, it is sufficient that any G out of the C generated combinations arrive at the receiver to be able to decode the original flits. The process of encoding and decoding has not even to be executed for the purpose of performance evaluation, since it just contributes an additional constant delay of a few clock cycles. This assumption accelerates the RLNC network simulation significantly. The RLNC code rate R is given as

$$R := \frac{G}{C}. \qquad (1)$$

Failure model: The failure model is used to mimic intermittent errors on selected ”hot spots”. A certain number of N error-prone routers is randomly selected in advance. Each of these routers has a constant probability f to drop a flit instead of forwarding it to the subsequent router (flit loss probability).

ARQ: If a flit loss is recognized at the receiver, an ARQ is sent back to the source to request a retransmission. Multiple flits might be requested at once for the uncoded case. No positive acknowledges are generated for saving bandwidth. The maximum number of retransmission trials is limited. The limit depends on the investigated scenario.

III. Simulation Setup

A cycle-accurate NoC traffic simulator is used for the generation of traffic commands and NoC topology, simulation and evaluation of the results. The simulator is based on a generic C++ framework that allows to arrange the routers in an arbitrary structure with different traffic schemes [13]. The network and traffic parameters are described in an XML format. An automated tool flow instantiates user-defined routers, links and network interfaces accordingly. The basic simulation framework was extended to consider error-prone routers with flit loss, RLNC transmissions, and ARQ. In our simulations, a scenario is always restricted to either uncoded or coded transmissions with fixed code rate. Mixed transmission modes are not investigated in the scope of this paper. In the following, some details on the most important implementation details are provided.

Error recognition: For the purpose of flit loss recognition, timers are implemented for each receiver. In case of coded traffic, two timers are used. When the first flit of a generation arrives at the receiver, a timer T1 is started to account for the interval between the arrival times of two successive flits of the same generation. In case of a timeout of T1, which indicates a flit loss, an ARQ is sent to the sender requesting a retransmission, and a larger timer T2 is started to account for the interval between sending ARQ and receiving retransmission.

The timeout of T2 also triggers an ARQ; in this case, either the ARQ or the retransmission was lost. The arrival of a new flit of the generation cancels timer T2 and starts timer T1 again. When the receiver has got G flits of a generation, it can start decoding and all timers related to this generation are deleted. Figure 2 shows a small example that demonstrates the interaction of RLNC transmission, ARQ, and timers.

An uncoded unicast with deterministic XY routing and ARQ serves as reference. In this case, error recognition is based on discontinuity in the flit IDs, and only timer T2 is used. Timer T2 is applied after sending an ARQ.

Regarding the selection of the timer values, an optimization is required. A small timer value would trigger too many ARQs and occupy bandwidth unnecessarily. A large timer value would cause large latency. The optimum value is found by a preceding design space exploration.

Retransmission Buffer: Before encoding, copies of the flits are saved in a circular buffer of the sender, so that retransmissions can be generated after receiving ARQs. Since old flits can be overwritten by new flits, the buffer size determines the number of retransmissions that can be

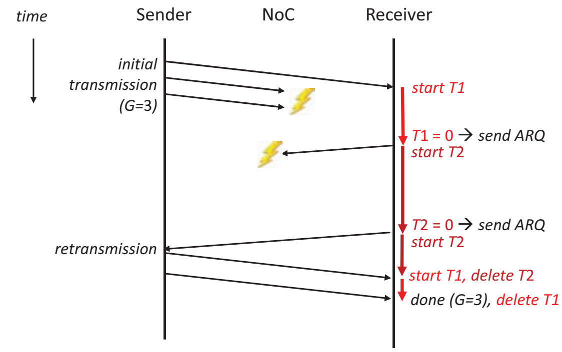


Fig. 2. Example for interaction of RLNC transmission, ARQ, and timers.

generated and thus affects latency and residual error rate. In the simulations a relatively large buffer size is chosen to avoid to effect of flit loss due to buffer overwriting. For practical realizations, the retransmission buffer must be dimensioned carefully. On the one hand, a large buffer allows multiple retransmission trials over long distance paths. On the other hand, the buffer size can affect the chip area of the NoC significantly.

Simulation parameters: In the simulations, an 8x8 2D mesh topology is studied. On average, 0.2 flits per router and per cycle are injected to the network (injection rate λ). However, the amount of effectively transmitted information λ_{eff} is influenced by the code rate R, due to the injection of redundant flits:

$$\lambda_{\text{eff}} := \lambda \cdot R. \tag{2}$$

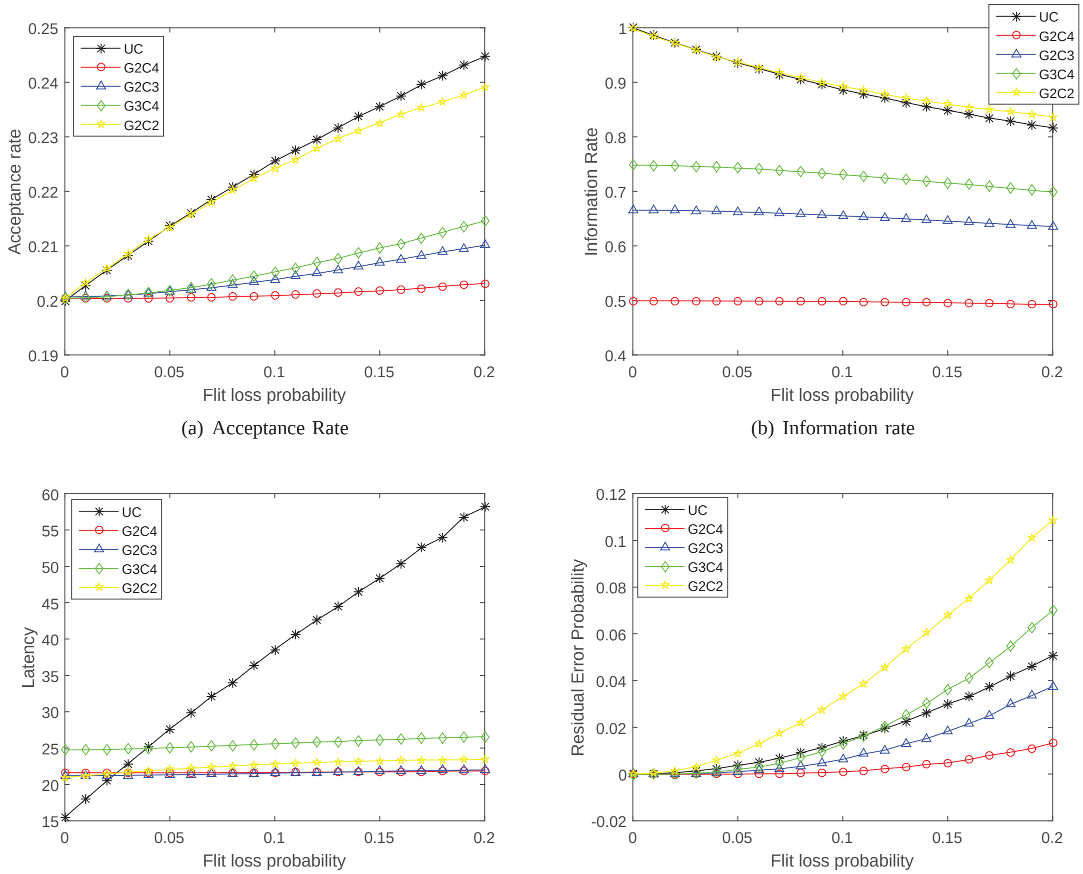
The following code rates are simulated: 2/2, 2/3, 2/4, and 3/4. We denote these cases as G2C2, G2C3, G2C4, and G3C4. The uncoded unicast (UC) serves as reference. Eight error-prone routers are randomly selected. For one simulation run, the flit loss probability f of all error-prone routers was set to a constant value. For every identified flit loss, at most one ARQ and one retransmission can be sent. Hence, a timeout of T2 does not trigger a further ARQ. Table I gives an overview of the simulation parameters.

TABLE I
Overview of simulation parameters.

Topology	2D mesh of size 8×8
Arbitration	Round-robin
Input buffers	Size: 4
Injection Rate	$\lambda = 0.2$
Code Rates	$R = G/C$ with [2/2, 2/3, 2/4, 3/4, UC]
Error-Prone Routers	8 (randomly selected)
Flit loss probability	$f = 0.01 \cdot i, i = 0, 1, \ldots, 20$
Max. # ARQ	1
Simulation run time	50k cycles

IV. Simulation Results

The results of the simulation are depicted in Fig. 3. Before we start the discussion of the results, a brief overview of the



(a) Acceptance Rate

(b) Information rate

(c) Average path latency

(d) Residual error probability

Fig. 3. Simulation results for 8x8 2D mesh with eight error-prone routers.

investigated performance metrics is given in the following.

A. Performance metrics

Four metrics have been chosen that are well suited for comparison and offer a good overview of different performance aspects in error-prone NoC.

ACCEPTANCE RATE A: The acceptance rate is the average number of flits per router and per cycle accepted by the network and represents the network load. Let N_{flits} be the total number of flits that are transmitted during the simulation run time, T_S the send time of the first flit, T_R the receive time of the last flit and M the total number of active modules. Then the acceptance rate can be given as follows

$$A := \frac{N_{\text{flits}}}{(T_R - T_S) \cdot M}. \qquad (3)$$

Without ARQ and retransmission, and under the assumption that the network is not saturated, the acceptance rate is equal to injection rate. The simulation results are shown in Fig. 3(a). It can be seen that the acceptance rate of uncoded transmissions exceeds those of the RLNC transmissions. This is due to the fact that for every discontinuity in the flit IDs an ARQ is generated. In contrast, the acceptance rate of RLNC coded transmissions is lower due to the included redundancy. The lower the code rate, the higher the redundancy and the lower the acceptance rate. With the increase of flit loss probability, more ARQs and retransmissions are sent and the difference between uncoded and coded transmissions becomes larger.

INFORMATION RATE I: The information rate represents the proportion of information flits and the total number of transmitted flits including redundant flits, ARQ flits (N_{arq}) and retransmissions (N_{retr}):

$$I := \frac{R \cdot (N_{\text{flits}} - N_{\text{arq}} - N_{\text{retr}})}{N_{\text{flits}}}. \qquad (4)$$

Figure 3(b) shows the simulation results for this metric. For a flit loss probability of zero, the information rate is equal to the code rate. With increasing number of ARQs and retransmissions in case of an increasing flit loss probability, the information rate decreases. For the case of uncoded transmissions, the information rate decreases more rapidly, because more ARQs and retransmissions are required.

LATENCY ℓ: The end-to-end path latency is the average time (in clock cycles) that a flit needs to travel from a sender to a receiver. For coded transmissions, it is equal to the number of cycles between sending the first flit of a generation and receiving the Gth flit of that generation. The results for the average path latency $\bar{\ell}$ are provided in Fig. 3(c). For lower flit loss probability, the latency of coded transmissions is higher than that of uncoded transmissions. This is because the router has to wait until G flits of the generation are received before the encoding resp. decoding can be started. This also explains the higher latency of G3C4 compared to other coded flits. With the increase of flit loss probability, the advantage of coded flits becomes more obvious. As illustrated in Fig. 1, RLNC effectively reduces the number of ARQs and retransmissions compared to uncoded transmissions. Therefore, it can reduce the average path latency significantly.

RESIDUAL ERROR PROBABILITY ϵ: The residual error probability is the proportion of flits which could not be received after the maximum allowed number of retransmission trials. The results for the average residual error probability $\bar{\epsilon}$ are shown in Fig. 3(d). The residual error decreases with the code rate and increases with flit loss probability. In the simulated setup, RLNC with code rates of G2C3 and G2C4 was able to achieve a lower residual error probability than uncoded transmissions.

B. Discussion

As it can be seen in Fig. 3, the main advantage of RLNC comes from its latency reduction in error-prone NoC (Fig. 3(c)). In the considered simulation scenario with eight error-prone routers, RLNC becomes already worthwhile for flit loss probabilities of $> 3\%$. It could be shown that a latency reduction of up to 62% is possible (for 20% flit loss, comparing the UC and G2C3/G2C4 curves). However, under good transmission conditions with low error rates, RLNC becomes disadvantageous due to the additional effort for encoding and decoding. For the zero flit loss case, we observed a latency increase of up to 66% (comparing the UC and G3C4 curves). Care must be taken when selecting the generation size. A high generation size requires a high initial cost w.r.t. the coding and decoding latency and is only worthwhile for large networks or under very poor transmission conditions.

As well as for the latency, RLNC also shows clear benefits concerning acceptance rate (Fig. 3(a)) and residual error probability (Fig. 3(d)). These two metrics follow the same trend: the lower the code rate, the more advantageous RLNC becomes compared to the UC transmissions. The residual error even shows a gain comparable to the latency improvement. It can be reduced by up to 66%, assuming a 20% flit loss probability. With respect to the network load, we could also observe a moderate reduction of the acceptance rate of up to 17% compared to the UC traffic case (for 20% flit loss probability).

Up to now, we were only discussing benefits of the RLNC performance under error-prone transmission conditions. Sure there must be some drawback as well. This is getting clear, if the information rate is considered as shown in Fig. 3(b). Due to the additional redundant flits (i.e. the generated random linear combinations) the information part is reduced significantly. However, analysing the curves more closely, it can be seen that the margin between UC traffic and RLNC traffic becomes smaller for higher error probability. Under the zero flit loss condition, a factor of two is observed comparing the UC and G2C4 curves. The gap is reduced to a factor of 1.65 at 20% flit loss.

We summarize that RLNC provides a flexible tool for resilient and low-latency transmission in error-prone NoC, which can even reduce network load. The advantages come at the cost of reduced information rate. The code rate parameter allows a flexible and dynamic adaptation of the transmission scheme according to the requirements and conditions. A preceding design space exploration is recommended to find the optimal schemes and parameter settings [14].

V. ANALYTIC MODEL

As the size of the network increases to accommodate thousands of cores on a chip, cycle accurate simulators require a prohibitive length of time to provide results. In this respect, analytic models become useful as they are faster and more flexible, provided that they can accurately model the system. Moreover, analytic models also provide an insight into the system which helps to explain the system behavior. In the current NoC setup without and with network coding, the four network performance parameters evaluated using the cycle accurate simulator are also evaluated by means of an analytic model that is introduced in the following. The common parameter symbols used in the model in addition to those introduced before are given in Tab. II.

TABLE II
OVERVIEW OF MODEL PARAMETER SYMBOLS

s	Source module
d	Destination module
$\lambda_{x,y}$	Flit injection rate from x to y
$\lambda'_{x,y}$	Total flit injection rate from x to y including ARQs and retransmission
$N_{x,y}$	Number of defect routers in the XY route from x to y
$f_{x,y}$	Total flit loss probability from x to y
$\bar{\epsilon}$	Average residual error probability
$\ell_{x,y}$	Path latency from x to y
$\ell'_{x,y}$	Total Path latency from x to y after retransmission
$\bar{\ell}$	Average path latency over all source-destination pairs

In the presence of the faulty routers with a certain flit loss probability f, the total flit loss probability $f_{s,d}$ between a source and destination pair depends on how many faulty routers ($N_{s,d}$) are present in this path from s to d:

$$f_{s,d} = 1 - (1 - f)^{N_{s,d}}. \quad (5)$$

The error probability is used in the calculation of the different network metrics for the uncoded and RLNC coded cases, as explained in the following sections.

A. Uncoded Network (UC)

ACCEPTANCE RATE: The injection and thus acceptance rate of the network increases with ARQ and retransmission. As explained in sections II and III, an error is recognized and an ARQ is triggered whenever there is a discontinuity in flit IDs of the received flits. The total injection rate $\lambda'_{s,d}$ is composed of the rate of the issued ARQs, $\lambda_{arq\_s,d}$ and the rate of the retransmitted flits, $\lambda_{retr\_s,d}$ in addition to the regular injection:

$$\lambda'_{s,d} = \lambda_{s,d} + \lambda_{arq\_s,d} + \lambda_{retr\_s,d}. \tag{6}$$

When computing $\lambda_{arq\_s,d}$ and $\lambda_{retr\_s,d}$, it has to be noted that an error is detected at the receiver d when 1 or more (e.g. t) consecutive flits from s to d are lost (with probability $f^t_{s,d}$) with the successive flit being received successfully (with probability $f'_{s,d} = 1 - f_{s,d}$). Since only a single ARQ is issued for all t consecutively lost flits, the rate of injection due to ARQs is proportional to $\frac{1}{t}$. Thus, $\lambda'_{arq\_s,d}$ is given by

$$\begin{aligned}\lambda_{arq\_s,d} &= \lambda_{d,s} \cdot f'_{d,s} \cdot \sum_{t=1}^{\infty} \frac{f^t_{d,s}}{t} \\ &= \lambda_{d,s} \cdot f'_{d,s} \cdot \ln \frac{1}{f'_{d,s}}.\end{aligned} \tag{7}$$

When an ARQ successfully reaches s (with probability $f'_{d,s}$), a retransmission is issued. Thus, $\lambda_{retr\_s,d}$ is given by

$$\begin{aligned}\lambda_{retr\_s,d} &= \lambda_{s,d} \cdot f'_{s,d} \cdot f'_{d,s} \cdot \sum_{t=1}^{\infty} f^t_{s,d} \\ &= \lambda_{s,d} \cdot f'_{d,s} \cdot f_{s,d}.\end{aligned} \tag{8}$$

Putting Eqs. 7 and 8 in Eq. 6 and taking the average over all modules, the acceptance rate is computed as follows:

$$A = \frac{1}{M} \cdot \sum_{s=1}^{M} \sum_{\substack{d=1 \\ d \neq s}}^{M} \lambda'_{s,d}. \tag{9}$$

INFORMATION RATE: Using the above mentioned injection rates and putting them in Eq. 4 with code rate $R = 1$, the information rate I is obtained as follows:

$$I = \frac{M(M-1)}{\sum_{s=1}^{M} \sum_{\substack{d=1 \\ d \neq s}}^{M} (1 + f'_{d,s} \cdot \ln \frac{1}{f'_{d,s}} + f'_{d,s} \cdot f_{s,d})}. \tag{10}$$

LATENCY: In the absence of faulty routers, the path latency $\ell_{s,d}$ is composed of the NoC interface injection and ejection delays and the total router transport delays. With retransmissions, the path latency now increases by the round trip delay, $\ell_{Rs,d}$ ($= 2 \cdot \ell_{s,d} + 2$) needed by ARQ and the retransmission to travel to and back from s. As the ARQ and retransmission must be first stored in a buffer and transmitted 1 cycle later, there is an additional delay of 2 cycles. Also included is the delay Δ ($=\frac{1}{\lambda_{s,d}}$) between consecutively injected flits for the same source-destination pair, as the ARQ is triggered only with the receipt of the next flit. Thus, the total path latency is obtained as shown in Eq. 11 by considering that t consecutive flits are lost before the next flit is received successfully with probability $f^t_{s,d} \cdot f'_{s,d}$. Moreover, the retransmission was received successfully with probability $f'_{d,s} \cdot f'_{s,d}$:

$$\begin{aligned}\ell'_{s,d} &= \ell_{s,d} \cdot f'_{s,d} + \sum_{t=1}^{\infty} (\Delta \cdot t + \ell_{Rs,d} + \ell_{s,d}) \cdot f^t_{s,d} \cdot {f'_{s,d}}^2 \cdot f'_{d,s} \\ &= \ell_{s,d} \cdot f'_{s,d} + [\Delta + f'_{s,d} \cdot (\ell_{Rs,d} + \ell_{s,d})] \cdot f_{s,d} \cdot f'_{d,s}.\end{aligned} \tag{11}$$

The average path latency is obtained by

$$\bar{\ell} = \frac{1}{M(M-1)} \sum_{s=1}^{M} \sum_{\substack{d=1 \\ d \neq s}}^{M} \ell'_{s,d}. \tag{12}$$

RESIDUAL ERROR PROBABILITY: The residual error probability is obtained by considering that even with retransmission, the flit fails to reach the destination as either the ARQ or the retransmission was lost. The average residual error probability of the system is given by:

$$\bar{\epsilon} = \frac{1}{M(M-1)} \sum_{s=1}^{M} \sum_{\substack{d=1 \\ d \neq s}}^{M} f_{s,d} \cdot (1 - f'_{d,s} \cdot f'_{s,d}). \tag{13}$$

B. RLNC coded Network (NC)

For the RLNC case, an error occurs if less than G flits of a generation are received. In this case, decoding of this generation is not possible. The error is recognized if at least one flit of the generation was received and there is a timeout of T1. The error probability is obtained using the binomial distribution, as in [15]:

$$f_{NCs,d} = \sum_{i=1}^{G-1} \binom{C}{i} {f'_{s,d}}^i f^{C-i}_{s,d}. \tag{14}$$

ACCEPTANCE RATE: The total injection rate per module is again composed of ARQs and retransmissions in addition to the regular injection rate $\lambda_{s,d}$. Since ARQs and retransmissions are sent per C combined flits of a generation, a factor $\frac{1}{C}$ has to be included in Eq. 15:

$$\lambda'_{NCs,d} = \lambda_{s,d} + \frac{\lambda_{d,s}}{C} \cdot f_{NCd,s} + \frac{\lambda_{s,d}}{C} \cdot f_{NCs,d} \cdot f'_{d,s}, \tag{15}$$

$$A_{NC} = \frac{1}{M} \sum_{s=1}^{M} \sum_{\substack{d=1 \\ d \neq s}}^{M} \lambda'_{NCs,d}. \tag{16}$$

INFORMATION RATE: Similiar to the uncoded case, the information rate is obtained by

$$I_{NC} = \frac{M(M-1) \cdot R}{\sum_{s=1}^{M} \sum_{\substack{d=1 \\ d \neq s}}^{M} (1 + \frac{f_{NCd,s}}{C} + \frac{f_{NCs,d}}{C} \cdot f'_{d,s})}. \tag{17}$$

LATENCY: In the absence of faulty routers, the sender first collects G flits of a generation, and then after computing C combinations, transmits them one after another. At the receiver, the combinations are collected and after the arrival of G flits, the flits are decoded and forwarded to the receiver module. Including theses delays in addition to buffering delays at the sender and the receiver, the path latency $\ell_{NCs,d}$ is obtained. With ARQ and retransmission, the total latency now includes the round-trip delay, $\ell_{Rs,d}$. The total and average latencies are given by Eqs. 18 and 19, respectively:

$$\ell'_{NCs,d} = \ell_{NCs,d} \cdot \sum_{i=G}^{C} \binom{C}{i} {f'_{s,d}}^{i} f_{s,d}^{C-i} + (\ell_{NCs,d} + \ell_{Rs,d}) \cdot \binom{C}{G-1} (f'_{s,d})^{G-1} {f_{s,d}}^{C-(G-1)} \cdot f'_{d,s} \cdot f'_{s,d}, \quad (18)$$

$$\bar{\ell}_{NC} = \frac{1}{M(M-1)} \sum_{s=1}^{M} \sum_{\substack{d=1 \\ d\neq s}}^{M} \ell'_{NCs,d}. \quad (19)$$

RESIDUAL ERROR PROBABILITY: As for the uncoded case, the residual error probability with retransmission increases with loss of the ARQ or the retransmitted flit. If $G - 2$ or less flits are received, decoding is not possible even with retransmission and so an error is also generated:

$$\bar{\epsilon}_{NC} = \frac{1}{M(M-1)} \sum_{s=1}^{M} \sum_{\substack{d=1 \\ d\neq s}}^{M} \Bigg[\sum_{i=0}^{G-2} \binom{C}{i} (1 - f_{s,d})^{i} {f_{s,d}}^{C-i} + \binom{C}{G-1} (f'_{s,d})^{G-1} {f_{s,d}}^{C-(G-1)} \cdot (1 - f'_{s,d} \cdot f'_{d,s}) \Bigg]. \quad (20)$$

C. Results and Discussion

The presented analytic model was used to evaluate the four main performance metrics of an 8x8 NoC having 8 faulty routers at the same location as in the simulator. All the results were found to closely match the simulator results. The maximum estimation error of the model relative to the simulation results are given in Tab. III. All parameters apart for the latency could be estimated by the analytic model with on average less than 1% error. The latency estimation by the model had a maximum error of 7% for the G2C2 case. For the other cases, the estimation error was equal to or less than 5%. All results have shown the same trend compared to the simulation results.

TABLE III
MAXIMATION ESTIMATION ERROR OF ANALYTIC MODEL

	UC (%)	G2C4(%)	G2C3(%)	G3C4(%)	G2C2(%)
A	1.26	0.34	0.47	0.82	0.28
I	1.22	0.34	0.3	0.89	0.27
$\bar{\ell}$	4.8	0.65	3.1	4.8	7.16
$\bar{\epsilon}$	0.77	1.4	1.3	1.8	0.04

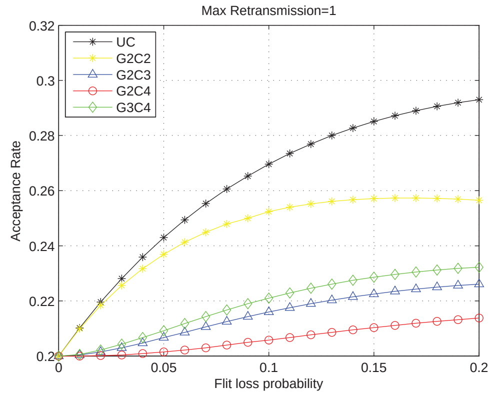
The benefit of using an analytic model is seen when evaluating large networks, in this case having 1024 cores. Such an evaluation using cycle accurate simulator would take days if not weeks. The large network was evaluated for the RLNC and UC case, having 128 error-prone routers, i.e., the same proportion as for the 8x8 NoC. The results are shown in Fig. 4. It can be observed that with increasing flit loss the high latencies of UC make it impractical for use in NOC. By using RLNC, compared to the UC case the latency can be reduced by 95% (at 20% flit loss, comparing the UC and RLNC curves). It has to be noted that a larger NoC has a higher average path length and as a result the probability of encountering a faulty router is greater. Thus the average error probability is higher for the large network. For the UC case, with retransmission the residual error rate is 10% higher at 3% flit loss (comparing 32x32 to 8x8 NoC). For RLNC with higher code rates (G2C2, G3C4), one retransmission per generation is not sufficient to complete the generation which makes decoding impossible and results in a higher residual error rate compared to UC. G2C3 does better in terms of residual error probability but only up to 13% flit loss compared to UC. For G2C4, the residual error probability is 25% less compared to UC (at 20% flit loss). When comparing the network load, RLNC performs better than UC, with a 27% lower acceptance rate for G2C4 (at 20% flit loss).

As large networks have higher residual error probabilities, RLNC can provide greater resilience but with a higher redundancy such as G2C4. In terms of network load and latency, which are two very important parameters for NoC, RLNC performs better than UC. As noted before in section IV-B, the advantage of RLNC comes at the price of reduced information rate. However, the gap in information rate between RLNC and UC becomes smaller at higher error probability. For the 32x32 NoC, when comparing UC and G2C4, a factor of 2 at zero flit loss is reduced to a factor of 1.46 at 20% flit loss.

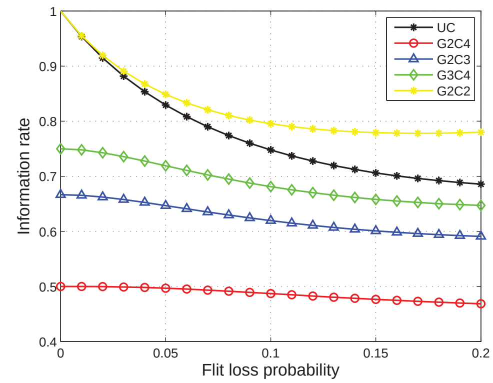
VI. CONCLUSION AND OUTLOOK

We presented an on-chip communication concept with random linear network coding for network-on-chip. In particular, we investigated the performance with a cycle-accurate simulator and provided an analytical model which produced very accurate performance estimations with much less computational complexity and allows for the investigation of large networks. Compared to the uncoded transmission scheme with ARQ and retransmission, the efficiency of RLNC is more obvious in larger networks and at higher flit loss probabilities. Thus, the use of network coding in many-core systems seems to be promising. The benefits concerning acceptance rate and residual error probability are observed especially at lower code rates. The drawback of RLNC is a lower information rate, which becomes less significant at higher flit loss probabilities.

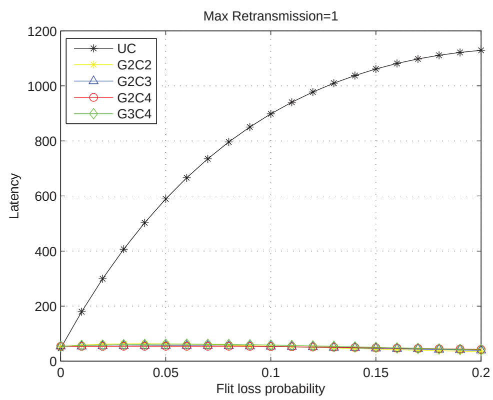
There are several interesting points up for future work. First, due to the assumption of uniform traffic, oblivious routing schemes like "Valiant's random routing" that allow for the utilization of spatial diversity and/or multiple paths were not included in this work. But it would be interesting to see their performance in application specific traffic patterns, especially w.r.t. resiliency and throughput. Second, we will analyze the efficiency if recoding at intermediate nodes is applied to further exploit the potential of RLNC. A third topic of future



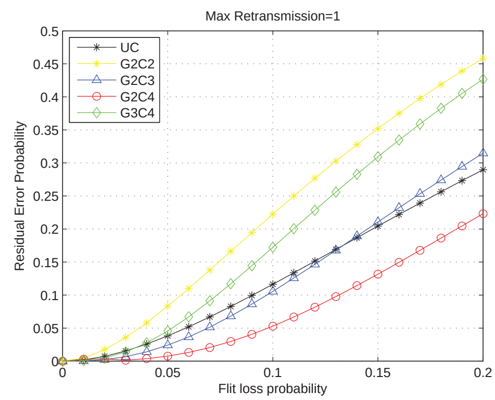
(a) Acceptance Rate



(b) Information Rate



(c) Average path latency



(d) Residual error probability

Fig. 4. Analytic model results for 32x32 2D mesh with 128 error-prone routers.

research is the investigation of multicast traffic with RLNC for NoC. Finally, we will also consider scenarios with more realistic assumptions w.r.t. faulty routers.

References

[1] W. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," in *Design Automation Conference, 2001. Proceedings*, 2001, pp. 684–689.

[2] B. Noethen *et al.*, "10.7 A 105GOPS 36mm2 heterogeneous SDR MPSoC with energy-aware dynamic scheduling and iterative detection-decoding for 4G in 65nm CMOS," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE Int.*, Feb 2014, pp. 188–189.

[3] S. Vangal *et al.*, "An 80-tile sub-100-w teraflops processor in 65-nm cmos," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 1, pp. 29–41, 2008.

[4] S. Borkar, "Thousand Core Chips: A Technology Perspective," in *Proceedings of the 44th Annual Design Automation Conference*, ser. DAC '07, 2007, pp. 746–749.

[5] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. on Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[6] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. of the IEEE International Symposium on Information Theory*, 2003.

[7] P. A. Chou, Y. Wu, and K. Jain, "Practical Network Coding," in *Proc. Annual Allerton Conf. on Comm., Control, and Computing*, 2003.

[8] L. S. Indrusiak, "Evaluating the feasibility of network coding for NoCs," in *Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, 2011, pp. 1–5.

[9] A. Shalaby, M. E.-S. Ragab, and V. Goulart, "Intermediate nodes selection schemes for Network Coding in Network-on-Chips," in *NORCHIP*, 2012, pp. 1–5.

[10] T. Duong-Ba, T. Nguyen, and P. Chiang, "Network Coding in Multicore Processors," in *IEEE 30th Int. Performance Computing and Communications Conference (IPCCC)*, 2011, pp. 1–7.

[11] M. Vonbun, S. Wallentowitz, M. Feilen, W. Stechele, and A. Herkersdorf, "Evaluation of Hop Count Advantages of Network-Coded 2D-Mesh NoCs," in *23rd Int. Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, 2013, pp. 134–141.

[12] W. J. Dally and B. Towles, "Principles and practices of interconnection networks," 2004.

[13] M. Winter and G. P. Fettweis, "A Network-on-Chip Channel Allocator for Run-Time Task Scheduling in Multi-Processor System-on-Chips," *Digital Systems Design, Euromicro Symp. on*, vol. 0, pp. 133–140, 2008.

[14] G. Ascia, V. Catania, A. G. D. Nuovo, M. Palesi, and D. Patti, "Efficient design space exploration for application specific systems-on-a-chip," *Journal of Systems Architecture*, vol. 53, no. 10, pp. 733 – 750, 2007.

[15] S. Pfennig and E. Franz, "Adjustable redundancy for secure network coding in a unicast scenario," in *Proc. International Symposium on Network Coding (NetCod)*, 2014.