Analysis of Application-Aware On-Chip Routing under Traffic Uncertainty

Nithin Michael, Milen Nikolov, Ao Tang, G. Edward Suh, Christopher Batten
School of Electrical and Computer Engineering, Cornell University, Ithaca, NY
{nm373,mvn22,at422,gs272,cbatten}@cornell.edu

ABSTRACT

Application-aware routing exploits static knowledge of an application's traffic pattern to improve performance compared to general-purpose routing algorithms. Unfortunately, traditional approaches to application-aware routing cannot efficiently handle dynamic changes in the traffic pattern limiting its usefulness in practice. In this paper, we study application-aware routing under traffic uncertainty. Our problem formulation allows an application to statically specify an uncertainty set of traffic patterns that each occur with a given probability, and our goal is to find a single set of combined routes that will enable high-performance across all of these traffic patterns. We show how efficient combined routes can be found for this problem using convex optimization. These combined routes are optimal when the performance for every traffic pattern using the combined routes is the same as the performance using routes that are specialized for just that traffic pattern. We derive necessary and sufficient conditions for when our optimization framework will find optimal combined routes. We use theoretical and numerical analysis for the important class of permutation traffic patterns to quantify how often optimal combined routes exist and to determine the performance loss when optimal combined routes are infeasible. Finally, we use a cycle-level simulator that includes realistic pipeline latencies, arbitration, and buffered flow-control to study the latency and throughput of combined routes compared to specialized routes and routes generated using general-purpose routing algorithms. The theoretical analysis, numerical analysis, and simulation results in this paper provide a first step towards more flexible application-aware routing.

Categories and Subject Descriptors

C.3 [**Peformance of Systems**]; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*routing protocols*

General Terms

Algorithms, Theory

NOCS'11, May 1–4, 2011, Pittsburgh, PA, USA.

1. INTRODUCTION

Most routing algorithms proposed for on-chip networks are general-purpose algorithms, since they are designed to perform well over a wide-range of applications. They are either completely oblivious to the application's traffic pattern (e.g., dimension ordered routing, ROMM [11], O1TURN [13]) or they dynamically adapt to an application's traffic pattern through indirect local information about the network's global performance (e.g., minimal adaptive routing [7], GOAL [17]). If the application's traffic pattern is known statically, then application-aware routing can potentially achieve better performance compared to general-purpose algorithms. In on-chip networks, application-aware routes are usually determined by either solving a mixed-integer linear program to generate single-path routes [9], or by solving the optimal-routing multicommodity flow problem [18] following earlier work done for general networks [2, 5]. We call this the *optimal specialized routing problem* since the goal is to find the optimal routes for a specific traffic pattern. These routes are then used to configure the on-chip network before executing the application. Single-path routes can use source-routing or table-based routing, while multi-path routes require table-based routing and per-router split-flow management. Multi-path routes offer increased network performance but also require more complicated hardware. Although application-aware multi-path routes are more common in wide-area-networks, they are an interesting research direction for on-chip networks and are the focus of this paper.

Unfortunately, application-aware routing cannot efficiently handle dynamic changes in the traffic pattern, and this is a serious limitation for modern workloads which often include many *application phases* [14, 15]. Each application phase exhibits significantly different behavior than other phases, and thus each application phase is characterized by a unique network traffic pattern. Application phases can last for thousands to millions of cycles. In this work, we assume that the phase traffic patterns are known statically, and that the sequence of application phases is either known statically or is determined dynamically at runtime. In other words, we have an uncertainty set of traffic patterns (one per application phase) each of which can occur with a given probability. The traffic patterns in the uncertainty set and the corresponding probabilities of occurrence are usually obtained through static analysis or by profiling the application of interest.

Given the uncertainty set, one approach to application-aware routing is to find the optimal network routes for each application phase, and then reconfigure the network at runtime before each phase. However, the cost of detecting application phases and reconfiguring the network can be high. We define the *combined optimal routing problem* as finding a single set of routes to be used across all application phases that results in the same performance or close to the same performance as if we used specialized routes for each application phase. Naive approaches to this problem include heuristically combining the optimal specialized routes for each application phase into a single set of routes, or combining the traffic patterns for all application phases into a single traffic pattern and solving a unified optimal-routing multi-commodity flow problem. We will show that neither of these naive approaches is optimal. Instead, we formulate the problem as a convex optimization problem (Section 2), and we use theoretical analysis (Section 3), numerical analysis (Section 4), and simulation results (Section 5) to illustrate that this formulation produces optimal solutions when possible and produces nearly-optimal solutions when the optimal solutions are infeasible.

2. OPTIMAL ROUTING UNDER TRAFFIC UNCERTAINTY

To put the problem that we are trying to solve in context and to introduce the notation used in the paper, we first describe the optimal routing problem for a single traffic pattern and its well-understood formulation as a convex optimization problem. We then discuss the more general combined optimal routing problem and illustrate how it can also be formulated as a similar convex optimization problem.

2.1 Specialized Optimal Routing Problem

An on-chip network interconnects terminals through a set of routers and unidirectional point-to-point channels (links). For this work, we focus on direct networks where there is one router per terminal, and we call the combination of a router and a terminal a node. Traffic patterns can be modeled by the communication between the different nodes. We denote the number of nodes in the network as N and the number of links as L. The capacities of the links are represented by $\mathbf{C} \in \mathbb{R}^L$. First, we define a few terms that will help us with the mathematical formulation of the problem.

Traffic Matrix/Pattern (D) **–** The traffic matrix $D \in \mathbb{R}^{N\times N}$ specifies the traffic requirements of the application. Each entry $D(s,d)$ represents the desired rate of data transfer from node s to node d and each such source-destination pair is said to constitute a network flow. We suppose that there are F non-zero flows in each traffic matrix, and we label the flow from s to d as the tuple $\langle s,d\rangle$.

Incidence Matrix (A) **–** The flow constraints imposed by the topology of the network are captured by its incidence matrix $A \in \mathbb{R}^{N\times L}$ which is defined as follows,

$$A(i,j) = \begin{cases} +1, & \text{if link } j \text{ is directed to node } i \\ -1, & \text{if link } j \text{ is directed away from node } i \\ 0, & \text{otherwise.} \end{cases}$$

Link Rates (Y) **–** $Y \in \mathbb{R}^{L\times F}$ represents the rate on each link due to each flow in the traffic matrix. It is easy to see that solving for the link rates for each flow specifies the route the flow takes through the network. We also define $\gamma = \sum_{j=1}^{F} Y^j$ as the vector of the total rate on each link required by the traffic matrix where Y^j represents the link rates corresponding to the flow j.

Cost Function (f) **–** We will use the following cost function.

$$f(\gamma) = \sum_{l=1}^{L} \frac{\gamma(l)}{C(l)-\gamma(l)} \tag{1}$$

With this formula, the cost function becomes the average number of packets in the system based on the hypothesis that each queue behaves as an M/M/1 queue of packets. Although this assumption is violated in real networks, the cost function described above provides a useful measure of performance in practice, because it expresses qualitatively the idea that congestion arises when the total rate on a link approaches its capacity as pointed out in [1]. Other measures of congestion include the maximum link utilization, but we do not consider them since a computational study has shown that the choice of the objective function between maximum link utilization and average number of packets in the network does not significantly impact the performance when used for routing optimization [19].

As noted earlier, we call the optimal routing problem for a single traffic matrix the *specialized optimal routing problem*, and it can be formulated as follows,

$$\begin{aligned} \underset{Y}{\text{minimize}} \quad & f(\gamma) \\ \text{subject to} \quad & AY = \overline{D}, \\ & \sum_{j=1}^{F} Y^j \le C, \\ & Y \ge 0. \end{aligned} \tag{2}$$

where the matrix $\overline{D} \in \mathbb{R}^{N\times F}$ is obtained from the traffic matrix D as follows,

$$\overline{D}(l,sd) = \begin{cases} +D(s,d), & \text{if } l = d \text{ for the flow } \langle s,d\rangle \\ -D(s,d), & \text{if } l = s \text{ for the flow } \langle s,d\rangle \\ 0, & \text{otherwise.} \end{cases}$$

The above formulation is a classic convex optimization problem and therefore can be solved efficiently to find the specialized optimal routes for the traffic matrix D.

2.2 Combined Optimal Routing Problem

Specialized routes are tuned for a single traffic pattern, but real applications often include a sequence of application phases each with their own traffic pattern. In other words, we have an uncertainty set $\mathbb{D} = \{D_1,\ldots,D_M\}$ of M traffic patterns that occur with probabilities $P_1,\ldots,P_M$. As noted earlier, we define the *combined optimal routing problem* as follows: find a single set of routes that enables the same performance or close to the same performance on each traffic pattern as if we used specialized routes for each traffic pattern. If the combined routes achieve the same performance as the specialized routes on each traffic pattern we call them *optimal combined routes*. Combined routes will enable us to configure the on-chip network once, and achieve optimal or near-optimal performance during all application phases. Fig. 1 illustrates the combined

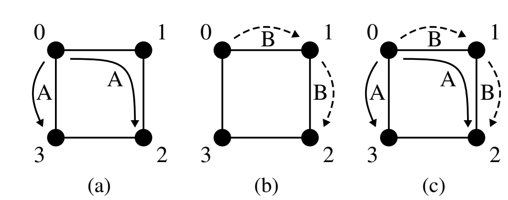


Figure 1: Specialized and Combined Routes for for Traffic Matrices D_A and D_B – $D_A(i,j) = D_B(i,j) = 0$ except for $D_A(0,3) = D_A(0,2) = D_B(0,1) = D_B(1,2) = 1$.

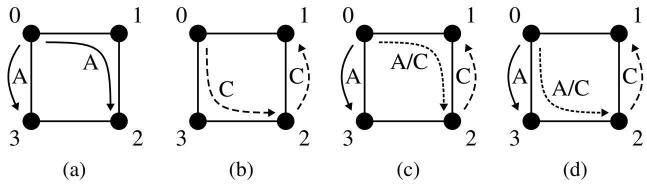


Figure 2: Specialized and Combined Routes for Traffic Matrices D_A and D_C – $D_A(i,j) = D_C(i,j) = 0$ except for $D_A(0,3) = D_A(0,2) = D_C(0,2) = D_C(2,1) = 1$.

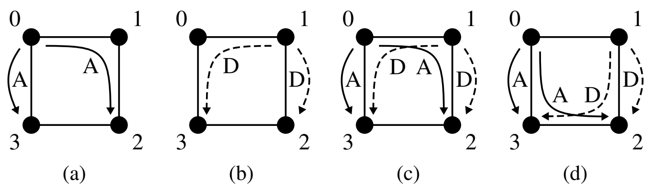


Figure 3: Specialized and Combined Routes for Traffic Matrices D_A and D_D – $D_A(i,j) = D_C(i,j) = 0$ except for $D_A(0,3) = D_A(0,2) = D_D(1,3) = D_D(1,2) = 1$.

optimal routing problem for a four-node ring network and two traffic patterns, D_A and D_B, each with two flows. Note that the example (and the other examples in this section) use single-path routing to simplify the discussion, but the illustrated concepts are common across both optimal single-path and multi-path routing. Fig. 1a and Fig. 1b illustrate optimal specialized routes for D_A and D_B respectively, and Fig. 1c illustrates combined routes for the uncertainty set containing D_A and D_B assuming each traffic pattern occurs with equal probability. Note that since the two traffic patterns have no flows in common, the optimal combined routes are simply the combination of the specialized routes for each traffic pattern. We can configure the four routers once with the routes show in Fig. 1c and we will achieve optimal performance during both application phases.

Given this example, a naive approach to the combined optimal routing problem is to simply solve the specialized optimal routing problem for each traffic matrix in the uncertainty set and then merge the resulting routes to create the combined routes. Unfortunately, this approach does not robustly handle flows that are shared across multiple traffic matrices. For example, Fig. 2a and Fig. 2b show the optimal specialized routes for two traffic patterns, D_A and D_C. The specialized route for flow $\langle 0,2\rangle$ is different for the two traffic patterns, and thus it is unclear which one to choose when combining the specialized routes. In Fig. 2c, we choose the specialized route from the solution to D_A, while in Fig. 2d, we choose the specialized route from the solution to D_C. The former results in optimal combined routes, while the latter will result in reduced performance when executing traffic pattern D_A since the link between node 0 to 3 is more heavily loaded. In larger network topologies with more complicated traffic patterns, it is not clear how to effectively combine optimal specialized routes. The key problem with this approach is that it considers the traffic matrices in the uncertainty set in isolation without considering their interaction when combining routes.

Another naive approach is to weight each element in each traffic matrix in the uncertainty set by the probability of that traffic matrix occurring and sum the corresponding elements across all traffic matrices. We can then solve the specialized optimal routing problem for this combined traffic matrix. Unfortunately, some solutions for the combined traffic matrix result in non-optimal combined routes. For example, Fig. 3a and Fig. 3b show the specialized routes for two traffic patterns, D_A and D_D. There are no shared flows, so combining D_A and D_D into a single traffic matrix is straightforward. Fig. 3c and Fig. 3d show two possible solutions to the specialized optimal routing problem for the combined traffic matrix. The former results in optimal combined routes, while the latter will result in reduced performance when executing traffic pattern D_A as well as D_D since the links connecting node 0 to node 3 and node 1 to node 2 respectively will be more heavily loaded. The key problem with this approach is that a single combined traffic matrix implies that flows from all traffic patterns in an uncertainty set happen simultaneously, but our problem formalation only uses a single traffic pattern during each application phase.

These examples illustrate that the key challenge in solving the combined optimal routing problem is creating a unified optimization framework that can determine *both* the specialized routes for each traffic pattern *and* the way these specialized routes interact to determine optimal combined routes. This is true for both single-path or multi-path routes. Our approach is to design an optimization problem that minimizes the expected cost function across all traffic matrices in the uncertainty set. In addition, the combined routes have to satisfy the requirements of every traffic matrix in the uncertainty set simultaneously. This means that if there is a flow that is shared across multiple traffic matrices, then the route computed for it should be the same for each of those traffic matrices. It is not immediately clear whether we can capture this intuitive constraint in terms of a convex constraint that will let us set up a convex optimization problem. Fortunately, the following theorem says that we can do precisely that.

Theorem 1. *Suppose that $\overline{D}_1$ and $\overline{D}_2$ both have a flow $\langle s,d\rangle$ and that $Y_1^{\langle s,d\rangle}$ and $Y_2^{\langle s,d\rangle}$ are the corresponding link rates. Then the route taken by the flow $\langle s,d\rangle$, as specified by the link rates, is the same for both traffic patterns if and only if $Y_1^{\langle s,d\rangle}/D_1(s,d) = Y_2^{\langle s,d\rangle}/D_2(s,d)$ where $D_i(s,d)$ represents the demand from s to d for traffic pattern D_i.*

Proof. Intuitively, the route for a flow $\langle s,d\rangle$ can be uniquely represented by how the flow splits at the intermediate nodes between the source and the destination. These split ratios indicate the route that one unit of traffic will take through the network from source to destination. We know that link rates can be uniquely determined from these node-based split ratios [5]. Since a route can be uniquely specified by node-based split ratios, it is easy to see that if the routes are the same between s and d for both $\overline{D}_1$ and $\overline{D}_2$ then the normalized rates on each link of the network will also be the same, i.e., $Y_1^{\langle s,d\rangle}/D_1(s,d) = Y_2^{\langle s,d\rangle}/D_2(s,d)$.

Now suppose that $Y_1^{\langle s,d\rangle}/D_1(s,d) = Y_2^{\langle s,d\rangle}/D_2(s,d)$. This implies that the normalized inflow into each node of the network and the normalized outgoing rates from each node are the same for both traffic patterns. So the split ratios at each node are the same for the flow between s and d for both $\overline{D}_1$ and $\overline{D}_2$. Since at each node, the split ratios corresponding to a flow completely define the route taken by the flow through the network, we can conclude that the route taken by flow $\langle s,d\rangle$ is the same for both traffic patterns. $\square$

Now we can define the optimal routing problem when there is

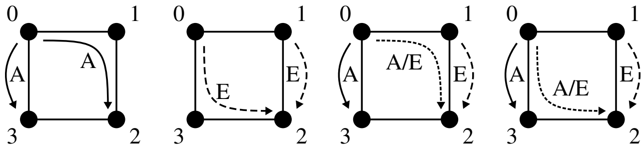


Figure 4: Specialized and Combined Routes for Traffic Matrices D_A and D_E – $D_A(i,j) = D_E(i,j) = 0$ except for $D_A(0,3) = D_A(0,2) = D_E(0,2) = D_E(1,2) = 1$.

uncertainty in the traffic pattern as follows,

$$
\begin{aligned}
\underset{\{Y_1,Y_2,\ldots,Y_M\}}{\text{minimize}} \quad & \sum_{i=1}^{M} p_i f(\gamma_i) \\
\text{subject to} \quad & FY_i = \overline{D}_i, i = 1,2,\ldots,M \\
& \sum_{j=1}^{F_i} Y_i^j \leq C, i = 1,2,\ldots,M \\
& Y_i \geq 0, i = 1,2,\ldots,M \\
& Y_i^{\langle s,d \rangle}/D_i(s,d) = Y_j^{\langle s,d \rangle}/D_j(s,d) \\
& \text{if flow } \langle s,d \rangle \text{ is in both } \overline{D}_i \text{ and } \overline{D}_j.
\end{aligned} \tag{3}
$$

Solving the above optimization problem is guaranteed to find the optimal combined routes if they exist. Suppose that there exists optimal combined routes for an uncertainty set $\mathbb{D} = \{D_1, \ldots, D_M\}$ and that the solution to Eq. 3 does not correspond to these routes. We can then show that a contradiction results, since selecting the optimal combined routes will further decrease the cost function in Eq. 3 as the $f(\gamma_i)$ corresponding to each D_i is minimized by the optimal combined routes by definition.

But there are still cases where it is simply not possible to find optimal combined routes. Fig. 4a and Fig. 4b illustrate the optimal specialized routes for two traffic matrices, D_A and D_E. Fig. 4c and Fig. 4d illustrate two solutions that are similar in spirit to what might be found using Eq. 3 (remember that for simplicity the examples in this section are for single-path routing, but our analysis is for general multi-path routing). Unfortunately, neither solution is optimal since both solutions result in more heavily loaded links compared to the optimal specialized routes. In this case, reconfiguring the network before each application phase would result in higher performance than using a single set of combined routes but perhaps this will not be necessary if the combined routes are close to optimal. The rest of the paper uses theoretical analysis, numerical analysis, and simulations to answer three key questions: (1) When are optimal combined routes feasible? (2) How often is this condition satisfied for different uncertainty set sizes? and (3) When optimal combined routes are not feasible, what is the performance loss compared to the optimal specialized routes?

3. THEORETICAL ANALYSIS

The effectiveness of the combined optimal routing framework is best characterized by how much performance we have to sacrifice for the additional flexibility that the framework offers. In this section, we first derive when we can find optimal combined routes, and we then study permutation traffic in more detail.

3.1 Necessary and Sufficient Conditions for Optimality

We present the necessary and sufficient condition obtained from the equilibrium conditions of Eq. 3 in the next theorem albeit with a simpler proof for ease of exposition. Identifying the particular cases when can achieve optimal combined routes and identifying the probabilities of their occurrence is an area that we are continuing to explore.

Theorem 2. *Suppose that $\overline{Y}_1, \ldots, \overline{Y}_M$ are the combined routes obtained from Eq. 3 and $Y_1^*, Y_2^*, \ldots, Y_M^*$ are the optimal routes corresponding to the traffic patterns $D_1, D_2, \ldots, D_M$. Then $f(\overline{\gamma}_i) = f(\gamma_i^*)$ $(i = 1, \ldots, M)$ if and only if $Y_i^{\langle s,d \rangle}/D_i(s,d) = Y_j^{\langle s,d \rangle *}/D_j(s,d)$ $(i, j = 1, \ldots, M)$ where $D_i(s,d) > 0$ and $D_j(s,d) > 0$ represents the demand from s to d for traffic patterns D_i and D_j respectively.*

Proof. Clearly if $Y_i^{\langle s,d \rangle *}/D_i(s,d) = Y_j^{\langle s,d \rangle *}/D_j(s,d)$ $(i, j = 1, \ldots, M)$ where $D_i(s,d) > 0$ and $D_j(s,d) > 0$ then the condition required to be satisfied by the shared flows in Eq. 3 is satisfied and consequently $Y_i^*(i = 1, \ldots, M)$ is a solution to Eq. 3 indicating that $f(\overline{\gamma}_i) = f(\gamma_i^*)$ $(i = 1, \ldots, M)$.

Next suppose that $f(\overline{\gamma}_i) = f(\gamma_i^*)$ $(i = 1, \ldots, M)$. Clearly $\overline{Y}_i(i = 1, \ldots, M)$ are optimal for the traffic matrices $D_1, D_2, \ldots, D_M$ respectively, i.e., $\overline{Y}_i = Y_i^*$ $(i = 1, \ldots, M)$. Also since $\overline{Y}_i(i = 1, \ldots, M)$ is a solution to Eq. 3 we know that the condition $\overline{Y}_i^{\langle s,d \rangle}/D_i(s,d) = \overline{Y}_j^{\langle s,d \rangle}/D_j(s,d)$ $(i, j = 1, \ldots, M)$ where $D_i(s,d) > 0$ and $D_j(s,d) > 0$ is satisfied for the shared flows between the pairs of traffic matrices D_i and D_j and so we conclude that $Y_i^{\langle s,d \rangle *}/D_i(s,d) = Y_j^{\langle s,d \rangle *}/D_j(s,d)$ $(i, j = 1, \ldots, M)$. $\square$

In words, the above result states that we can find optimal combined routes for a given uncertainty set, if and only if for any shared flow there exists optimal routes obtained by solving Eq. 2 that are the same for all the traffic matrices in the uncertainty set that contain that flow.

But this condition as stated is difficult to verify in practice and we would like to find verifiable conditions. For instance, from the structure of Eq. 3, it is clear that if there are no shared flows between the D_is then the problem decouples into M independent optimal routing problems and the routes obtained by solving Eq. 3 will be individually optimal for each element of $\mathbb{D}$. This gives us an easy to check sufficient condition for when solutions to Eq. 3 are optimal combined routes.

3.2 Permutation Traffic Matrices

The analysis in the previous section applies to all traffic patterns, but in this section we narrow our focus to just permutation traffic matrices. In these traffic patterns, each row and each column has only one non-zero entry. For a network with N nodes there are $N!$ possible permutation traffic matrices. First, for the sake of simplicity, suppose that the uncertainty set $\mathbb{D}$ consists of two permutation traffic matrices. The next theorem tells us how likely it is that these two traffic matrices do not have any shared flows.

Proposition 1. *The number of traffic patterns that do not share a flow with a given permutation traffic pattern for a network that has N nodes is given by $P_N = N! - \sum_{i=0}^{N-1} \binom{N}{i} P_i$ where $P_0 = 1$. Furthermore, we see that $\lim_{N\to\infty} P_N/N! = 1/e$ or that as $N \to \infty$ the probability of selecting a pair of permutation traffic matrices which do not share flows tends to $1/e$.*

Proof. We first note that for the purpose of determining if there are shared flows between two traffic matrices, the rates required by the flows do not matter. So we index flows simply by a 1 if a flow exists and a 0 otherwise. Any permutation matrix can be converted to any other permutation matrix by left multiplying it with a suitable permutation matrix. We note that a permutation traffic pattern A will share a flow with another permutation traffic pattern B only if the permutation matrix that transforms A's permutation matrix into the permutation matrix of B has non-zero diagonal entries. By eliminating all permutation matrices with non-zero diagonal entries from the set of permutation matrices leaves us with the transformations that will yield traffic patterns that do not share a flow with a given permutation traffic pattern. Consequently we have $P_N = N! - \sum_{i=0}^{N-1} \binom{N}{i} P_i$. Here we set $P_0 = 1$ for brevity in notation.

The second part of the theorem is similar to the famous "Hat Check Problem" studied by Bernoulli and Montmort although we provide a different solution. We have,

$$\sum_{i=0}^{N} \binom{N}{i} P_i = N!$$
$$\Rightarrow \sum_{i=0}^{N} \binom{N}{i} k_i i! = N!$$
$$\Rightarrow \sum_{i=0}^{N} k_i/(N-i)! = 1$$

Using induction we can show that $k_i = \sum_{j=0}^{i} (-1)^j/j!$. First we note that for $N = 0$, $k_0 = 1$. Then applying the induction hypothesis to $\sum_{i=0}^{N} k_i/(N-i)!$ yields

$$\sum_{i=0}^{N} \sum_{j=0}^{i} \frac{(-1)^j}{j!(N-i)!}$$
$$= \sum_{k=0}^{N} \sum_{j=0}^{k} \frac{(-1)^j}{j!(k-j)!}$$
$$= \sum_{k=0}^{N} 0^k/k!$$
$$= 1$$

Since $k_i = \sum_{j=0}^{i} (-1)^j/j!$, we have $\lim_{i\to\infty} k_i = 1/e$ completing the proof. □

As one might expect this probability decreases as the number of traffic matrices in $\mathbb{D}$ increases. Another interesting restriction is obtained when we study what the maximum size of the set $\mathbb{D}$ can be if we consider only permutation traffic patterns that do not share flows.

Proposition 2. *The cardinality of $\mathbb{D}$ is N if we restrict attention to permutation traffic patterns that do not share flows.*

Proof. If the traffic patterns do not share flows, by the pigeon hole principle we can conclude that the cardinality of $\mathbb{D}$ can be at most N. To see that it is indeed N, we observe that row rotating an $N\times N$ identity matrix yields a set of N permutation matrices with corresponding traffic patterns that do not share flows. Multiplying any given permutation matrix by this set yields a set of permutation matrices corresponding to traffic patterns that do not share flows. □

M	% of all with $\Gamma = 1$	% of opt with shared	% of non-opt with $\Gamma < 1.05$	max Γ	% of non-opt with given num shared flows: 1	2	3	4	5+
2	65	62	100	1.004	29	57	14	0	0
3	20	72	100	1.009	29	18	12	18	23
4	10	100	87	1.073	0	0	16	5	79
5	3	100	82	1.090	0	0	0	0	100
6	0	–	78	1.150	0	0	0	0	100

Table 1: Results of Numerical Analysis – Columns list the size of the uncertainty sets (M); percentage of all uncertainty sets with loss factor (l) of one (i.e., optimal combined routes are feasible); percentage of optimal combined routes for which there is at least one shared flow; percentage of non-optimal combined routes with a loss factor less than 1.05; maximum loss factor over all uncertainty sets; percentage of non-optimal combined routes with the given number of shared flows.

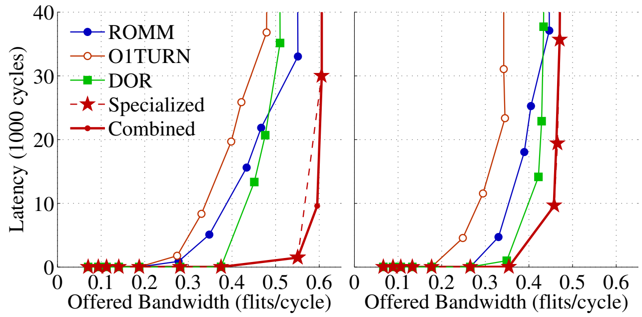
For uncertainty sets comprising well-structured permutation traffic patterns, with the above results we are able to characterize to some extent when we have optimal combined routes. However, as pointed out earlier, even in this case it is challenging to determine every situation in which we can find optimal combined routes and if there is loss in optimality to quantify the loss. Consequently, we rely on numerical experiments to help us further characterize the performance of the combined optimal routing framework.

4. NUMERICAL ANALYSIS

In this section, we empirically answer, as the size of the uncertainty set increases, how often we can find optimal combined routes and if optimal combined routes do not exist, what is the *loss factor*. The loss factor (Γ) is the factor by which the average number of packets with combined optimal routes differ from that with optimal specialized routes. In other words, $\Gamma = \sum_{i=1}^{M} p_i f(\overline{\gamma}_i)/\sum_{i=1}^{M} p_i f(\gamma_i^*)$. Once again, we restrict attention to permutation traffic patterns in order to obtain a more complete characterization of the performance of the combined optimal routing framework on this important class of traffic patterns.

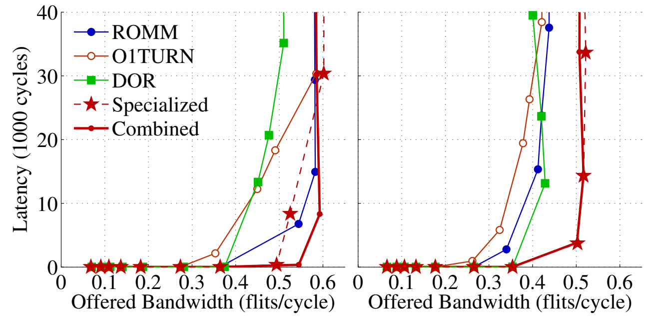
We performed our evaluations over uncertainty sets with two to six traffic matrices on a 6×6 two-dimensional mesh. For each set size, we randomly generated 500 uncertainty sets and solved the corresponding specialized and combined optimal routing problems with the objective being to minimize the average number of packets in the network. Note that these numerical experiments involve solving multiple convex optimization problems with several hundred thousand variables. Even though the optimization problems were solved efficiently using cvx [6], the calculations for each uncertainty set took on the order of hours to complete and the complete numerical analysis required many thousands of hours of computation. The results from the numerical analysis are summarized in Table 1.

The first metric that we studied was the empirical probability of being able to find optimal combined routes. In order to go beyond the analytical results of the previous section in quantifying the performance of the combined optimal routing framework, we also studied the probability of finding optimal combined routes when the traffic matrices in the uncertainty set shared flows. But as the size of the uncertainty set was increased, the empirical probability



(a) (b)

Figure 5: Latency vs. Offered Bandwidth for Optimal Combined Routes – Theoretical and numerical analysis predicts the combined routes should be able to acehive the same throughput as the specialized routes.



(a) (b)

Figure 6: Latency vs. Offered Bandwidth for Sub-Optimal Combined Routes – Theoretical analysis shows that optimal combined routes are infeasible, but numerical analysis predicts that the combined routes should still perform close to the specialized routes.

of finding optimal combined routes decreased as we would expect. Encouragingly, for most of the uncertainty sets that were generated, combined optimal routes performed to within 5% of the optimal specialized routes as can be seen.

This observation naturally led to the next question which was when we did lose optimality, how bad was the loss? For uncertainty set sizes 2 and 3, at least in our sample sets, very low loss in optimality was observed. However, for larger uncertainty set sizes, fairly high values of Γ were observed in the worst case. But even in these cases, we expect that the combined optimal routes will perform better than the general-purpose routing algorithms as the next section will illustrate. For the uncertainty sets with non-optimal combined routes, we also studied the percentages of occurrence of different numbers of shared flows. The idea was to study the correlation between the number of shared flows and the probability of an uncertainty set having an undesirably high value for Γ.

5. SIMULATION RESULTS

The simulator that we used was DARSIM [10], a cycle-level on-chip network simulator. All the simulations were performed on a 6×6 two-dimensional mesh network. The simulator was given a warm-up period of 20,000 cycles after which performance statistics were collected over 100,000 cycles in order to ensure the accuracy of the results. The primary performance criteria that we measured were throughput and latency. The data rates are expressed in flits/cycle and each packet is divided into 8 flits. Also the simulator was configured so that each physical channel was divided into 6 virtual channels with 8 flits of buffering each. The capacity of the physical channel was set to be 1 flit/cycle. In the simulator, virtual channels are pre-allocated to the different flows once the routes are computed so that deadlock is avoided according to the static virtual channel allocation scheme described in [16].

The aim of the simulations was to get an idea of how factors like buffering and flow control influenced the performance of the optimal routes. We conducted our study with two uncertainty sets of size two where one had optimal combined routes and the other did not. In order to compare the performance of the optimal routing scheme with the general-purpose routing algorithms, we also studied how ROMM, O1TURN, and DOR performed for the traffic matrices in the uncertainty sets. As expected, from the latency-throughput curves in Fig. 5, the optimal combined routes match the peak throughput achieved by the optimal specialized routes for both traffic matrices in the uncertainty set while outperforming the general-purpose routing algorithms.

More interesting results can be observed from the uncertainty set with non-optimal combined routes. From the previous section we expect that for uncertainty set size two, the combined routes should perform very close to the optimal specialized routes. Even factoring in the affects of non-idealities introduced by the simulator, we see from Fig. 6 that the specialized and combined routes are very close to each other in performance. Once again, it can be observed that the application-aware schemes yield better performance than the general-purpose algorithms.

Of course it would be interesting to continue exploring the space of permutation matrices and study how the non-ideal characteristics of on-chip networks affect the performance of combined optimal routing on larger uncertainty sets where adversarial traffic matrices can result in larger loss factors. The simulation and numerical results suggest this and many other directions of continued research to give us a better understanding of the properties of the combined optimal routing framework which appears to be a promising first step towards introducing optimal routing with a certain degree of flexibility to networks on-chip.

6. RELATED WORK

In the context of on-chip networks, application-aware optimal single-path routing for a single traffic pattern was explored in [3,9]. But the focus on single path routes made the optimal routing problem NP-hard and consequently inefficient to solve. On the other hand, optimal multi-path routing for a single traffic pattern was explored even earlier [18]. Unlike the optimal single-path routing problem, the optimal multi-path routing problem is convex and therefore can be solved efficiently to determine optimal routes. Our work computes optimal multi-path routes as well, but differs from the previous work in that we are computing the routes for an uncertainty set of traffic patterns.

Another approach to application-aware routing can be found in [12] where the idea is to map the application's communication graph to the network in such a way as to avoid cycles in the channel dependency graph. Then minimal adaptive deadlock-free routing is performed on the acyclic application-aware channel dependency graph. However, here knowledge of the application is just used to avoid deadlock by a suitable mapping of the traffic requirements to the network. It is not used to try and optimize the performance of the routing algorithm with respect to any metric.

The problem of dealing with traffic uncertainty when formulating the optimal routing problem has only begun to receive attention over the last few years. Algorithms like COPE [20] approach the problem by trying to minimize the worst case performance of the routing scheme within an uncertainty set. The problem of finding optimal routes by minimizing the expected cost over a set of traffic patterns has been studied previously in the context of intra-domain routing in the Internet [21]. However, the focus was on setting up the problem and extending the results of [5] to develop a distributed solution method. Our work goes further by providing conditions for the existence of optimal combined routes, empirically studying the probability that these conditions are met, and quantifying the loss in optimality when optimal combined routes do not exist.

7. CONCLUSIONS

The paper presents a first step towards more flexible application-aware routing in on-chip networks. In order to get around the fact that specialized optimal routing is not viable for traffic patterns other than the one that it was designed for, we introduce combined optimal routing that can handle uncertainties in the traffic patterns that might be produced in the on-chip network. The performance of the schemes and their advantages are characterized analytically, numerically, and through simulations. Importantly, we derive necessary and sufficient conditions for the combined routes to have no loss in optimality. Numerical experiments with randomly generated uncertainty sets of permutation traffic matrices provide empirical evidence that combined optimal routing can perform very close to specialized optimal routing. The simulation results obtained using sample points from these randomly generated uncertainty sets show that this observation from numerical experiments holds even with realistic pipeline latencies, arbitration, and buffered flow-control. Overall, the initial results indicate that the combined optimal routing framework is a promising technique that adds flexibility to application-aware optimal routing.

There are a number of future directions that we plan to pursue in order to further our understanding of the structure of the combined routing problem and address practical implementation issues. For instance, we would like to better quantify how likely optimal combined routes exist for more complex traffic patterns. We are also currently exploring how to bound the loss in optimality in the combined routing problem. The simulation results imply that the loss is typically small, but analytical results will provide a better understanding given that simulations alone cannot the sweep the entire parameter space.

There also exist practical implementation issues that need further investigation. For example, multi-path routes can potentially cause out-or-order delivery of flits and require buffers to re-order them. It is also necessary to have a good deadlock scheme in order to fully exploit the performance advantages of optimal routing. In the paper, we used static allocation of virtual channels in order to avoid deadlock [16]. We plan to further study different deadlock avoidance schemes such as resource ordering [8] or escape channels [4], and their performance implications. Also, we believe that it is important to couple the optimal routing schemes with a suitable flow control scheme in order to fully exploit their performance advantages. Consequently, identifying a good flow control scheme is of particular interest to us. Finally, in order to be able to use the routes generated by the techniques described in the paper, we need a router that is capable of splitting flows. Designing and evaluating the router and ways to encode the routes efficiently are two other directions for future research.

ACKNOWLEDGMENTS

This work was funded in part by the National Science Foundation (Awards CNS-0708788, CCF-0835706 and CCF-0905208), a Jacobs Graduate Research Fellowship (Michael), and an equipment donation from Intel Corporation. The authors acknowledge and thank Francis Ciaramello for his help running simulations on the Cornell Teragrid.

REFERENCES

[1] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, 1992.

[2] D. Cantor and M. Gerla. Optimal Routing in a Packet-Switched Computer Network. *IEEE Trans. on Computers*, 23(10):1062–1069, 1974.

[3] M. Cho et al. Diastolic Arrays: Throughput-Driven Reconfigurable Computing. *Int'l Conf. on Computer-Aided Design*, Nov 2008.

[4] J. Duato. A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks. *IEEE Trans. on Parallel and Distributed Systems*, 4(12):1320–1331, Dec 1993.

[5] R. Gallager. A Minimum Delay Routing Algorithm Using Distributed Computation. *IEEE Transactions on Communications*, 25(1):73–85, Jan 1977.

[6] M. Grant and S. Boyd. CVX: Matlab Software for Disciplined Convex Programming, version 1.21. `http://cvxr.com/cvx`, Feb 2011.

[7] L. Gravano et al. Adaptive Deadlock- and Livelock-Free Routing with All Minimal Paths in Torus Networks. *IEEE Trans. on Parallel and Distributed Systems*, 5(12):1233–1252, Dec 1993.

[8] K. Gunther. Prevention of Deadlocks in Packet-Switched Data Transport Systems. *IEEE Trans. on Computers*, 29(4):512–524, Apr 1981.

[9] M. Kinsy et al. Application-Aware Deadlock-Free Oblivious Routing. *Int'l Symp. on Computer Architecture*, Jun 2009.

[10] M. Lis et al. DARSIM: A Parallel Cycle-Level NoC Simulator. *Workshop on Modeling, Benchmarking, and Simulation*, Jun 2010.

[11] T. Nesson and S. Johnsson. ROMM Routing on Mesh and Torus Networks. *Symp. on Parallelism in Algorithms and Architecture*, Jul 1995.

[12] M. Palesi et al. Application Specific Routing Algorithms for Networks on Chip. *IEEE Trans. on Parallel and Distributed Systems*, 20:316–330, Mar 2009.

[13] D. Seo et al. Near-Optimal Worst-Case Throughput Routing for Two-Dimensional Mesh Networks. *Int'l Symp. on Computer Architecture*, Jun 2005.

[14] X. Shen et al. Locality Phase Prediction. *Int'l Conf. on Architectural Support for Programming Languages and Operating Systems*, Dec 2004.

[15] T. Sherwood et al. Automatically Characterizing Large Scale Program Behavior. *Int'l Conf. on Architectural Support for Programming Languages and Operating Systems*, Oct 2002.

[16] K. Shim et al. Static Virtual Channel Allocation in Oblivious Routing. *Int'l Symp. on Networks-on-Chips*, May 2009.

[17] A. Singh et al. GOAL: A Load-Balanced Adaptive Routing Algorithm for Torus Networks. *Int'l Symp. on Computer Architecture*, Jun 2003.

[18] B. Towles et al. Throughput-Centric Routing Algorithm Design. *Symp. on Parallelism in Algorithms and Architecture*, Jun 2003.

[19] K. Vastola. A Numerical Study of Two Measures of Delay for Network Routing. Technical report, University of Illinois, Sep 1979.

[20] H. Wang et al. COPE: Traffic Engineering in Dynamic Networks. *SIGCOMM*, Sep 2006.

[21] C. Zhang et al. On Optimal Routing with Multiple Traffic Matrices. *INFOCOM*, Mar 2005.