

doi: 10.17586/2226-1494-2024-24-5-687-698  
УДК 004.3

## Обзор алгоритмов маршрутизации для сетей на кристалле

Михаил Игоревич Бондаренко<sup>1</sup>, Алексей Евгеньевич Платунов<sup>2</sup>✉

<sup>1,2</sup> Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация

<sup>1</sup> ООО «Синтакор», Санкт-Петербург, 196247, Российская Федерация

<sup>1</sup> [bondarenko-m-i@yandex.ru](mailto:bondarenko-m-i@yandex.ru), <https://orcid.org/0009-0002-6816-4055>

<sup>2</sup> [aeplatunov@itmo.ru](mailto:aeplatunov@itmo.ru) ✉, <https://orcid.org/0000-0003-3003-3949>

### Аннотация

**Введение.** В работе исследованы алгоритмы маршрутизации для сетей на кристалле (СенК). Приведен анализ существующих алгоритмов маршрутизации, выделены их ограничения и области применения. Оценка алгоритмов проведена с учетом требований конкретных приложений и особенностей архитектур. Представлены результаты сравнения производительности рассматриваемых алгоритмов. **Метод.** Анализ и сравнение различных алгоритмов маршрутизации для СенК проведены с учетом критически важных характеристик. Основное внимание уделено таким алгоритмам маршрутизации, как детерминированный XY-алгоритм, алгоритм поворота модели, маршрутизация с учетом перегрузок, отказоустойчивая маршрутизация, маршрутизация с учетом «качества услуги», алгоритм муравьиной колонии. Показано, что выбор алгоритма маршрутизации должен основываться на специфических требованиях и условиях использования сети. Показана важность адаптации к разнообразным условиям и задачам, с которыми могут столкнуться пользователи и разработчики СенК. **Основные результаты.** С использованием данных из существующих исследований проведен анализ алгоритмов на основе нескольких ключевых показателей: задержка, пропускная способность, адаптивность, отказоустойчивость и сложность реализации. Выявлены сильные и слабые стороны каждого алгоритма в различных сценариях использования и при разной нагрузке на сеть. Показано, что выбор алгоритма маршрутизации должен опираться на конкретные требования и условия использования сети, а также на баланс между производительностью, адаптивностью, отказоустойчивостью и сложностью реализации. **Обсуждение.** Исследование вносит вклад в понимание эффективности различных алгоритмов маршрутизации в СенК. Предоставлены рекомендации для их выбора в зависимости от специфических требований приложения и архитектуры системы. Исследование способствует углублению понимания влияния алгоритмов маршрутизации на общую эффективность СенК. Предложены направления дальнейших усовершенствований в этой области. Результаты работы могут быть применены при проектировании и разработке высокопроизводительных многопроцессорных систем на кристалле, где эффективная маршрутизация данных между различными компонентами системы является ключевым фактором обеспечения высокой производительности. Подчеркнута значимость разработки отказоустойчивых алгоритмов маршрутизации, способных обеспечивать непрерывность работы системы в случае отказов отдельных компонентов или узлов. Это особенно важно для критических приложений, где непрерывность сервиса и снижение риска потери данных являются приоритетными задачами.

### Ключевые слова

интегральные схемы, многоядерные системы, сети на кристалле, шаблоны трафика, взаимоблокировка, ресурсное голодание, алгоритмы маршрутизации, детерминированные алгоритмы маршрутизации, отказоустойчивая маршрутизация, маршрутизация с учетом перегрузок, эффективность сети

**Ссылка для цитирования:** Бондаренко М.И., Платунов А.Е. Обзор алгоритмов маршрутизации для сетей на кристалле // Научно-технический вестник информационных технологий, механики и оптики. 2024. Т. 24, № 5. С. 687–698. doi: 10.17586/2226-1494-2024-24-5-687-698

## Overview of routing algorithms for network on chip

Mikhail I. Bondarenko<sup>1</sup>, Alexey E. Platunov<sup>2</sup>✉

<sup>1,2</sup> ITMO University, Saint Petersburg, 197101, Russian Federation

<sup>1</sup> ООО “Syntacore” Saint Petersburg, 196247, Russian Federation

<sup>1,2</sup> bondarenko-m-i@yandex.ru, <https://orcid.org/0009-0002-6816-4055>

<sup>2</sup> aeplatunov@itmo.ru✉, <https://orcid.org/0000-0003-3003-3949>

### Abstract

This paper examines routing algorithms for networks on a chip (NoC). An analysis of existing routing algorithms is provided; their limitations and areas of application are highlighted. The algorithms were evaluated taking into account the requirements of specific applications and architecture features. The results of comparing the performance of the considered algorithms are presented. Analysis and comparison of various routing algorithms for NoC are carried out taking into account critical characteristics. The main attention is paid to such routing algorithms as the deterministic XY algorithm, the model rotation algorithm, congestion-aware routing, fault-tolerant routing, Quality of Service routing, and the ant colony algorithm. It is shown that the choice of routing algorithm should be based on the specific requirements and conditions of use of the network. The importance of adapting to a variety of conditions and tasks that NoC users and developers may encounter is shown. Based on data from existing studies, an analysis of algorithms was carried out based on several key indicators: latency, throughput, adaptability, fault tolerance and implementation complexity. The strengths and weaknesses of each algorithm are identified in various use scenarios and under different network loads. It is shown that the choice of a routing algorithm should be based on the specific requirements and conditions of use of the network, as well as on the balance between performance, adaptability, fault tolerance and implementation complexity. The study contributes to the understanding of the effectiveness of various routing algorithms in NoC, providing recommendations for their selection depending on the specific application requirements and system architecture. The study contributes to a better understanding of the impact of routing algorithms on the overall performance of NoC, suggesting directions for further improvements in this area. The results of the work can be applied in the design and development of high-performance multiprocessor systems on a chip where efficient data routing between various systems components is a key factor in ensuring high performance. The importance of developing fault-tolerant routing algorithms that can ensure the continuity of system operation in the event of failures of individual components or units is emphasized. This is especially important for mission-critical applications where service continuity and reducing the risk of data loss are top priorities.

### Keywords

integrated circuits, multicore systems, network on a chip, traffic patterns, deadlock, resource starvation, routing algorithms, deterministic routing algorithms, fault-tolerant routing, congestion-aware routing, network efficiency

**For citation:** Bondarenko M.I., Platunov A.E. Overview of routing algorithms for network on chip. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2024, vol. 24, no. 5, pp. 687–698 (in Russian). doi: 10.17586/2226-1494-2024-24-5-687-698

### Введение

В настоящее время сети на кристалле (СенК) признаются ключевым компонентом в архитектуре высокопроизводительных и многопроцессорных систем на кристалле (СнК) [1]. Из-за возрастающих требований к производительности и энергоэффективности в современных вычислительных системах эффективность маршрутизации в СенК становится критически важной [2]. Однако сложность маршрутизации усугубляется разнообразием топологий сети, динамикой нагрузки и ограничениями по задержке и пропускной способности [3]. Это влечет за собой необходимость разработки алгоритмов маршрутизации, способных оптимизировать эти параметры, учитывая специфические требования приложений и архитектуры системы [4–7].

В настоящей работе рассмотрена проблема маршрутизации в СенК, ее влияние на общую производительность и энергоэффективность СнК. Выполнен анализ существующих подходов к маршрутизации в СенК, описаны их преимущества и недостатки.

### Предмет исследования

Архитектура СнК постоянно меняется, чтобы соответствовать современным тенденциям производительности и энергопотребления. Многоядерная архитектура

в настоящее время активно используется для многих СнК, и эта тенденция приобретает популярность по мере того, как все больше разработчиков выбирают этот подход. Производительность СнК в таком случае будет определяться архитектурой межсоединений. При использовании коммуникационных архитектур на основе шины снижаются временные характеристики (особенно это проявляется на этапе физического проектирования) и в целом разработка становится сложной.

Современные подходы к проектированию многоядерных СнК в качестве архитектуры межсоединений используют СенК. В такой методологии проектирования применены концепции вычислительных сетей и крупномасштабных параллельных вычислительных систем. СенК представляет собой структурированную связь между блоками и преодолевает узкие места шины с точки зрения масштабируемости.

СенК — инновационный подход к проектированию коммуникационных систем внутри интегральных схем, особенно в контексте многопроцессорных СнК. Они обеспечивают масштабируемую и эффективную методику для управления взаимосвязью между различными модулями, такими как процессоры, память и специализированные ускорители [8, 9]. Маршрутизация становится актуальной, когда число подключаемых компонентов на чипе оказывается достаточно большим, чтобы прямое их соединение стало невозможным или

неэффективным с точки зрения площади чипа и потребления энергии. При этом необходимость в реализации специальных процессов маршрутизации начинается с чипов, содержащих от 4 до 8 ядер. Современные СнК могут включать десятки или даже сотни ядер, и ожидается, что в будущем их число будет расти, учитывая тренды к увеличению производительности и масштабированию. СенК может занимать значительную часть площади чипа, особенно в сложных СнК с высокой степенью интеграции и большим числом компонентов. Это относится к реализации как коммутационных элементов и каналов связи, так и к логике для управления маршрутизацией и обработкой ошибок. Удельный вес сети в общей площади чипа может изменяться и для сложных многоядерных процессоров составлять от 10 до 30 %. Временные задержки, связанные с передачей данных через СенК, важны для общей производительности СнК. Задержки могут отличаться в зависимости

от длины пути, загруженности сети и эффективности механизмов маршрутизации. В то время как современные технологии стремятся минимизировать эти задержки, они все еще играют важную роль в определении производительности высокопроизводительных вычислительных систем.

Архитектура СенК основана на принципах, используемых в компьютерных сетях и телекоммуникациях, и адаптирована для масштаба интегральных схем. Основные компоненты СенК включают вычислительные узлы, маршрутизаторы для пересылки данных между узлами и каналы связи, соединяющие маршрутизаторы. Архитектура может быть организована в различных топологиях, таких как сетка, дерево, кольцо или нестандартные топологии, специфичные для требований конкретной системы [10–12].

На рис. 1 показаны типичные архитектуры СнК на основе шинной и сетевой архитектур.

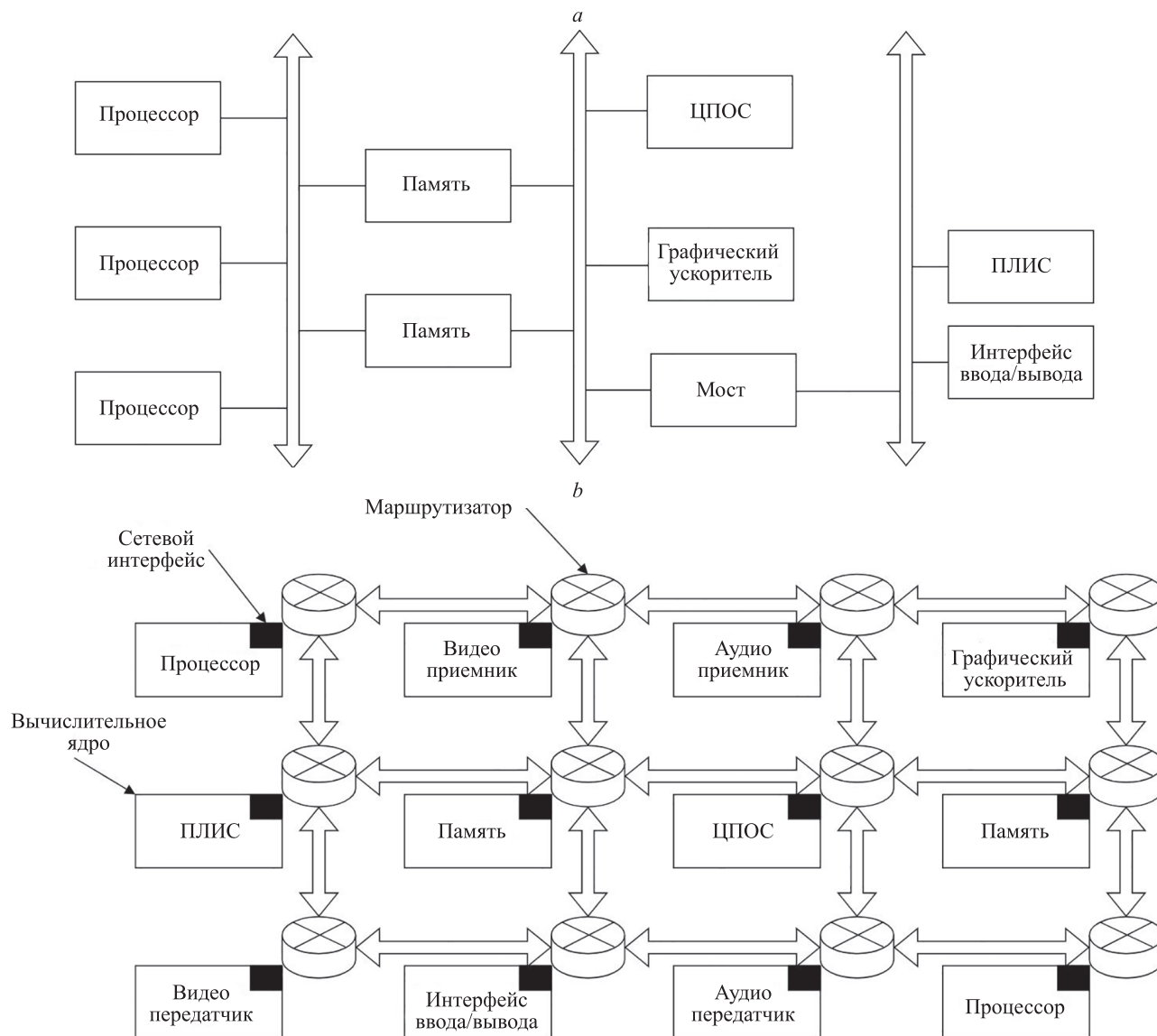


Рис. 1. Пример систем на кристалле на основе шинной (а) и сетевой архитектур (б). ЦПОС — цифровой процессор обработки сигналов, ПЛИС — программируемая логическая интегральная схема

Fig. 1. Example of systems on a chip based on bus (a) and network (b) architectures. ЦПОС — digital signal processor, ПЛИС — programmable logic integrated circuit

Сетевая архитектура имеет преимущества по сравнению с традиционными шинными СнК. Например, она обеспечивает высокую масштабируемость, позволяя эффективно увеличивать количество узлов без значительного ухудшения производительности, что является критическим для комплексных многопроцессорных СнК. В отличие от шин, где передача данных ограничена одним каналом, СенК поддерживает множество параллельных путей передачи, уменьшая задержки и увеличивая пропускную способность. Также архитектура СенК позволяет оптимизировать коммуникацию в соответствии с конкретными требованиями приложения, включая оптимизацию энергопотребления и снижение задержек.

СенК отличаются от других типов сетей рядом уникальных характеристик, обусловленных их применением в интегральных схемах. Так, СенК физически интегрированы в один кристалл полупроводника и предназначены для использования в микроэлектронике. В то время как другие типы сетей могут использовать разнообразные физические среды передачи, включая оптоволокно и радиочастоты. Кроме того, СенК предназначены для очень малых физических размеров, обычно в пределах одного кремниевого кристалла. Это принципиально отличает их от других сетей, которые могут охватывать от нескольких метров до тысяч километров. Благодаря близости компонентов СнК друг к другу и использованию специализированных протоколов и топологий, СенК могут обеспечивать очень высокую пропускную способность и минимальную задержку. Для других классов сетей обычно характерны большие задержки и ограниченная пропускная способность по причине больших расстояний и использования более универсальных протоколов. В СенК важно минимизировать энергопотребление из-за ограничений тепловыделения в интегральных схемах. В других сетях, хотя энергоэффективность также является важным фактором, ограничения по тепловыделению могут быть менее критичны. Также в СенК, как и в любой другой интегральной полупроводниковой вычислительной системе, обеспечение отказоустойчивости является критически важным вследствие отсутствия возможности внести исправления в готовый кристалл. В других типах сетей различные вышедшие из строя компоненты можно заменить исправными.

Таким образом, СенК представляют собой специализированный тип сети, оптимизированный для коммуникации внутри микроэлектронных устройств, что существенно отличает их от других сетевых технологий, предназначенных для более широкого спектра применений и условий эксплуатации.

В современном мире вычислительных систем и сетевых технологий, алгоритмы маршрутизации играют ключевую роль в обеспечении эффективности и надежности передачи данных между различными узлами в системе. Во время использования алгоритмов необходимо учитывать возможные проблемы. Например, важно равномерно распределять трафик по сети для оптимизации использования ресурсов и предотвращения узких мест, что требует сложных механизмов управления потоками и буферизации. Также мини-

мизация энергопотребления при сохранении высокой производительности маршрутизации является важной задачей, особенно в мобильных и встраиваемых системах. Различные типы сетевого трафика в СенК могут различаться в зависимости от приложения и архитектуры конкретной СнК [13, 14]. Общие модели включают в себя трафики: равномерный (каждый узел взаимодействует с любым другим узлом с равной вероятностью), локализованный (связь между определенными узлами происходит чаще [15, 16]) и трафик с всплесками (характеризуется внезапным всплеском требований к связи [17]). Существуют различные виды блокировок и недостатки ресурсов. Например, взаимоблокировка (Deadlock) возникает, когда набор пакетов блокируется навсегда, при этом каждый пакет ожидает ресурсов, которые удерживаются другими пакетами. Это создает циклическую зависимость, при которой ни один пакет не может перемещаться, что фактически останавливает связь в сети [18, 19]. Ситуация, когда пакеты постоянно меняют свое состояние или положение, но никогда не продвигаются к месту назначения, называется живой блокировкой. В отличие от взаимоблокировки, пакеты не останавливаются, вместо этого они постоянно двигаются, но без продуктивных результатов [20]. Один из разновидностей недостатка ресурсов – ресурсное голодание, которое происходит, когда определенные пакеты в сети постоянно задерживаются или им отказывают в доступе к ресурсам, в то время как другие пакеты продолжают обрабатываться нормально. Это приводит к неопределенной отсрочке некоторых пакетов [21–23].

## Методы и материалы

Каждый алгоритм маршрутизации может характеризоваться несколькими показателями. В данной работе в качестве основных метрик для сравнения алгоритмов были выбраны несколько ключевых показателей — производительность (совокупность задержки и пропускной способности), адаптивность, отказоустойчивость и сложность реализации. Рассмотрим подробнее каждый из критериев сравнения (табл. 1).

В рамках настоящей работы рассмотрены различные типы статистических и динамических алгоритмов, а также дана их краткая характеристика по перечисленным критериям сравнения. Анализ основан на данных из существующих исследований, что позволяет выявить сильные и слабые стороны каждого алгоритма в различных сценариях использования и при разной нагрузке на сеть.

Дадим характеристику каждому алгоритму по перечисленным критериям.

**XУ-алгоритм.** Детерминированный алгоритм, в котором пакеты сначала маршрутизируются по оси  $X$ , а затем по оси  $Y$ . Данный алгоритм обычно обеспечивает низкую задержку и высокую пропускную способность в условиях равномерного трафика благодаря своей простоте и предсказуемости. Адаптивность отсутствует, так как алгоритм не реагирует на изменения в сети, такие как перегрузка или сбои. Отказоустойчивость низкая, поскольку она не может динамически менять

Таблица 1. Метрики для сравнения алгоритмов  
Table 1. Metrics for comparing algorithms

Критерий сравнения	Описание критерия
Задержка	Критически важный критерий, влияющий на производительность сети, особенно в приложениях реального времени. Исследования показывают, что алгоритмы, использующие предварительные вычисления маршрутов, как правило, обеспечивают меньшую задержку по сравнению с алгоритмами динамической маршрутизации. Однако, в условиях высокой нагрузки на сеть, динамические алгоритмы способны более эффективно адаптироваться к изменениям, что потенциально может снизить задержку за счет оптимизации путей передачи данных
Пропускная способность	Данный критерий зависит от эффективности используемых алгоритмов маршрутизации. В контексте пропускной способности алгоритмы, способные оптимизировать использование сетевых ресурсов и минимизировать коллизии, демонстрируют лучшие результаты. Выполненные исследования доказали, что алгоритмы с поддержкой множественных путей могут значительно увеличить пропускную способность за счет распределения трафика по нескольким маршрутам
Адаптивность	Адаптивность алгоритмов маршрутизации позволяет сетям эффективно реагировать на изменения в трафике и топологии сети, таким образом оптимизируя производительность сети в динамичных условиях. Алгоритмы, способные к самообучению и самонастройке, обеспечивают высокую адаптивность, что позволяет им эффективно решать задачи маршрутизации при изменяющихся условиях, включая перегрузку сети и изменение требований к задержке и пропускной способности
Отказоустойчивость	Учитывается способность алгоритма маршрутизации поддерживать непрерывность работы сети в случае отказов отдельных компонентов или узлов. Алгоритмы, обладающие высокой отказоустойчивостью, способны быстро восстанавливать работоспособность сети, перенаправляя трафик через альтернативные маршруты. Это обеспечивает непрерывность сервиса и снижает риск потери данных. Важный фактор — способность алгоритма предвидеть и компенсировать потенциальные точки отказа, а также его способность к быстрой реорганизации маршрутов в реальном времени
Сложность реализации	Сложность реализации алгоритма маршрутизации включает в себя как программную, так и аппаратную составляющие. Этот критерий сравнения оказывает значительное влияние на выбор алгоритма для конкретной вычислительной системы или сети. Алгоритмы с высокой сложностью реализации требуют более значительных ресурсов для разработки и тестирования, а также могут потребовать специализированного аппаратного обеспечения для оптимальной работы. Кроме того, более сложные алгоритмы могут представлять вызовы в плане обслуживания и масштабирования

маршрут вокруг неисправных узлов. Сложность очень низкая, что упрощает реализацию и понимание.

Работы [24, 25] представляют обзоры детерминированных алгоритмов маршрутизации, уделяя особое внимание XY-алгоритму, который является одним из наиболее простых и предсказуемых. Однако его основное ограничение — неспособность адаптироваться к перегрузкам и сбоям узлов, что делает его менее эффективным в условиях высокодинамичных и ненадежных сетей. В работах [24, 25] систематизированы знания о детерминированных алгоритмах маршрутизации, подробно рассмотрены применимость и ограничения данного алгоритма в различных сценариях использования. Для дальнейшего развития направления необходимы гибридные алгоритмы, сочетающие простоту и предсказуемость XY-алгоритма с адаптивностью к перегрузкам и отказам. Также актуально исследование динамических методов маршрутизации, способных более эффективно реагировать на изменения в условиях сети.

**Алгоритм поворота модели.** Данный алгоритм ограничивает определенные повороты, чтобы избежать тупиков. В эту категорию попадают такие варианты, как «West First», «North Last» и «Negative First». Алгоритм обеспечивает лучшую производительность, чем XY-алгоритм, благодаря частичной адаптивности,

что помогает избегать определенных перегрузок сети. Отметим, что алгоритм поворота модели не является полностью динамичным. Отказоустойчивость у него лучше, чем у XY-алгоритма, но не так надежна, как у полностью адаптивных или отказоустойчивых алгоритмов. Сложность выше, чем у XY-алгоритма, но управляема.

Работы [26, 27] оказали большое влияние на развитие алгоритмов маршрутизации для СенК, предлагая адаптивные модели маршрутизации для повышения производительности и надежности сети. Алгоритм поворота модели предлагает улучшенную адаптивность за счет возможности обхода перегруженных и неисправных узлов, что улучшает общую производительность сети. Повышается отказоустойчивость сети, позволяя динамически менять маршруты и обеспечивать непрерывность работы при сбоях. Увеличение пропускной способности и снижение задержек достигается за счет оптимизации маршрутов и динамического распределения трафика. Увеличение сложности алгоритмов требует значительных вычислительных ресурсов, что может ограничить их применение в масштабируемых системах. Вопросы энергоэффективности остаются недостаточно исследованными, что важно для современных многопроцессорных систем. Необходимы дополнительные эксперименты и симуляции для под-

тверждения эффективности предложенных решений в реальных условиях. В работах [26, 27] предложены перспективные решения для маршрутизации в СенК, однако они требуют дальнейших исследований для уменьшения сложности реализации и улучшения энергоэффективности. Эти аспекты могут стать объектом будущих исследований, направленных на оптимизацию производительности и надежности СенК.

**Маршрутизация с учетом перегрузок.** Алгоритм специально разработан для обнаружения и предотвращения перегруженных областей в сети. Решения по маршрутизации принимаются на основе текущей нагрузки трафика с целью его балансировки и уменьшения узких мест [28, 29]. Уменьшение задержки и повышение пропускной способности достигается за счет исключения перегруженных маршрутов. Данная маршрутизация имеет умеренную отказоустойчивость, так как в первую очередь ориентирована на перегрузку. В зависимости от механизма обнаружения и управления перегрузками сложность варьируется от средней до высокой.

В работе [28] рассмотрен алгоритм, направленный на уменьшение задержек и повышение пропускной способности. Однако его тестирование проводилось только на синтетических нагрузках, что вызывает сомнения в его эффективности в реальных условиях. Кроме того, не рассмотрены вопросы масштабируемости алгоритма при увеличении размеров сети. Работа [29] описывает гибридный алгоритм маршрутизации, сочетающий детерминированные и адаптивные методы для повышения устойчивости сети к перегрузкам. Несмотря на то, что экспериментальные данные подтверждают его эффективность, в [29] недостаточно освещены вычислительные затраты, связанные с реализацией гибридного подхода. Это требует дальнейшего исследования для оценки влияния на производительность и оптимизации алгоритма. Таким образом, работы [28, 29] имеют ценные идеи, но требуют дополнительной проверки и оптимизации для практического применения в больших и сложных СенК.

**Отказоустойчивая маршрутизация.** При использовании данного алгоритма основное внимание уделяется обеспечению непрерывной работы при наличии неисправностей в сети. Алгоритм динамически идентифицирует и обходит неисправные компоненты. Может наблюдаться увеличение задержки из-за обхода неисправных мест, он показывает высокую эффективность при адаптации к сбоям в сети, в меньшей степени при перегрузках. Высокая отказоустойчивость сети подтверждается. Сложность изменяется, но обычно увеличивается из-за необходимости обнаружения и устранения неисправностей.

В работах [30, 31] предложено повышение отказоустойчивости в СенК. Описан динамичный алгоритм, эффективно обходящий неисправные участки сети, обеспечивая непрерывность сервиса. Также проведен формальный анализ подобного алгоритма, подтверждая его корректность и надежность. Основным преимуществом динамического алгоритма является высокая адаптивность к отказам и возможность поддержания работоспособности сети в критических условиях. Тем

не менее, такой алгоритм может столкнуться с проблемами, связанными с повышенной задержкой при обходе неисправностей и высокой сложностью реализации, что требует значительных вычислительных ресурсов. Таким образом, данные исследования содержат полезные сведения об отказоустойчивой маршрутизации в СенК, но требуют дальнейших исследований для оптимизации алгоритмов и снижения их сложности.

**Маршрутизация с учетом качества обслуживания (Quality of Service, QoS).** В результате работы данного алгоритма выполняется обработка различных типов трафика, гарантируя, что пакеты с высоким приоритетом получают более быстрые и надежные маршруты. Алгоритм может оптимизировать задержку или пропускную способность на основе приоритетов пакетов и адаптируется к условиям сети на основе требований QoS. Отказоустойчивость варьируется, так как не является основной задачей этого алгоритма. Сложность высокая, поскольку требует сложных механизмов для категоризации и определения приоритетности трафика.

Работы [32, 33] посвящены улучшению QoS в СенК через разработку специализированных алгоритмов маршрутизации. Предложена методика управления потоками QoS, которая способна адаптироваться к различным условиям сети и обеспечивать гарантии по задержке и пропускной способности. Однако работа [32] не предоставляет достаточно данных о производительности в условиях высоких нагрузок и не рассматривает аспекты отказоустойчивости. Кроме того, неясна масштабируемость предложенного подхода при увеличении числа узлов. Также рассмотрен алгоритм, основанный на поведении пчел, для адаптивной маршрутизации трафика с учетом QoS. Этот инновационный подход требует значительных вычислительных ресурсов и сложен в реализации. В [33] отсутствует сравнительный анализ с существующими алгоритмами, а также детальные данные об отказоустойчивости. Требуются дальнейшие исследования по следующим направлениям — анализ производительности в условиях высоких нагрузок, оценка отказоустойчивости и масштабируемости предложенных методов, сравнение с существующими алгоритмами для объективной оценки эффективности.

**Алгоритм муравьиной колонии.** Алгоритм основан на поведении муравьев при поиске путей к источникам пищи. Он использует искусственных муравьев для исследования сети и прокладки феромонных следов (т. е. присвоение весов). Решения о маршрутизации основаны на этих уровнях феромонов, которые отражают полезность путей. Задержка и пропускная способность могут быть эффективными, но изначально хуже на этапе обучения. Адаптивность алгоритма высокая — происходит постоянная оптимизация путей в зависимости от условий сети. Отказоустойчивость хорошая, благодаря своей динамичности и способности находить новые пути. Сложность довольно высокая, из-за затрат на управление виртуальными муравьями и следами феромонов.

В [34, 35] рассмотрены современные подходы к маршрутизации СенК, включая алгоритм муравьиной колонии. Работа [34] анализирует производительность

и адаптивность — алгоритм демонстрирует хорошие показатели по задержке и пропускной способности, особенно после начального этапа обучения. Он эффективно адаптируется к изменениям в сети, что позволяет ему быстро оптимизировать маршруты. Способность алгоритма находить новые пути обеспечивает высокую отказоустойчивость системы. Вместе с тем у данного алгоритма существуют нерешенные проблемы. Например, высокие затраты на управление виртуальными муравьями и следами феромонов могут вызвать сложности при внедрении. Также на этапе обучения алгоритм может демонстрировать повышенную задержку, что критично для систем с высокими требованиями к начальной производительности. Работа [35] включает детальный анализ производительности, адаптивности, отказоустойчивости и сложности реализации различных алгоритмов маршрутизации. Это дает всестороннее представление о текущем состоянии исследований и ценные рекомендации для разработчиков. Недостаток примеров внедрения описанных алгоритмов в реальных системах снижает их прикладную ценность. Работа охватывает широкий спектр алгоритмов, но глубина анализа каждого из них могла бы быть более подробной.

Чтобы проиллюстрировать отличительные особенности в алгоритмах маршрутизации между СенК и другими категориями сетей, рассмотрим пример алгоритма маршрутизации ХУ, используемого в СенК, и сравним его с дистанционно-векторным алгоритмом маршрутизации, обычно применяемого в более крупных компьютерных сетях.

Приведем особенности ХУ-алгоритма СенК, позволяющие улучшить ХУ-маршрутизацию. Сетчатая топология СенК позволяет ХУ-маршрутизации быть высокоэффективной и предсказуемой, обеспечивая минимальное отклонение от маршрута и оптимизируя задержку. Избегая сложных вычислений пути, ХУ-алгоритм минимизирует задержку маршрутизации и поддерживает высокую пропускную способность, необходимую для внутрикристалльных коммуникаций. Простота ХУ-алгоритма делает его ресурсосберегающим, требующим меньше энергии и площади, чем более сложные алгоритмы, что имеет решающее значение в условиях ограниченных возможностей чипа.

Маршрутизация по дистанционно-векторному алгоритму используется в крупномасштабных сетях, таких как Интернет. Он предполагает, что маршрутизаторы совместно используют свои таблицы маршрутизации с ближайшими соседями и каждый маршрутизатор обновляет таблицу маршрутизации на основе полученной информации, чтобы определить кратчайший путь к каждому пункту назначения. Среди различий для маршрутизации по дистанционно-векторному алгоритму можно выделить адаптивность к динамическим топологиям — в отличие от ХУ-маршрутизации. Данный алгоритм может адаптироваться к изменениям в топологии сети, таким как сбой соединения и добавление новых узлов, что делает его подходящим для больших и динамических сетей. Также пути, определяемые дистанционно-векторным алгоритмом, могут изменяться с течением времени по мере изменения топологии сети, в отличие от предопределенной топо-

логии в СенК. Необходимость периодического обмена таблицами маршрутизации и вычисления путей может привести к дополнительной задержке и потреблению большей полосы пропускания (полосу пропускания — объем данных, который может быть передан через сеть за определенный период времени). В контексте СенК, полоса пропускания зависит от нескольких ключевых факторов: архитектура сети, трафик и выбранный алгоритм маршрутизации.

Использование СенК может привести к улучшениям алгоритмов, например, оптимизация для конкретной топологии — известная топология СенК позволяет разрабатывать алгоритмы маршрутизации, которые непосредственно адаптированы к архитектуре сети, повышая эффективность и предсказуемость. Также алгоритмы маршрутизации СенК могут быть оптимизированы наряду с общей архитектурой чипа, что позволяет использовать целостный подход, учитывающий как вычислительные, так и коммуникационные требования, оптимизируя энергопотребление, задержку и пропускную способность. Существующие возможности настройки алгоритмов маршрутизации под конкретные требования приложений или схемы трафика в СенК могут значительно улучшить показатели производительности, в отличие от более общих подходов, требуемых в крупных и гетерогенных сетях.

Таким образом, рассмотренный пример показал, как уникальные характеристики СенК позволяют разрабатывать высокоэффективные и оптимизированные алгоритмы маршрутизации, адаптированные к конкретным требованиям СенК.

## Результаты и обсуждение

В рамках работы был выполнен анализ, по результатам которого можно дать оценку эффективности алгоритмов и показать в каких обстоятельствах каждый алгоритм проявит себя лучше всего. Алгоритмы оценены по трехбалльной шкале (один закрашенный квадратик соответствует 1 баллу), результаты сведены в табл. 2.

Из табл. 2 видно, что для простых и предсказуемых СенК ХУ-маршрутизация эффективна и достаточна [36, 37]. В сценариях с умеренной изменчивостью и перегруженностью маршрутизация с использованием модели поворота является сбалансированным выбором [38, 39]. При работе с изменяющейся нагрузкой трафика оптимальным является маршрутизация с учетом перегрузки [40]. В системах, подверженных сбоям, отказоустойчивая маршрутизация имеет решающее значение для надежности [41, 42]. Для разнообразных требований к трафику QoS маршрутизация предлагает индивидуальные решения [43, 44]. В высокодинамичных и сложных системах алгоритм муравьиной колонии предлагает инновационную адаптивность, основанную на обучении [45].

На представленных графиках (рис. 2) показана ожидаемая зависимость задержки и пропускной способности для различных алгоритмов маршрутизации при разных уровнях нагрузки на сеть, которую планируется подтвердить в дальнейшем после получения результатов симуляции.

Таблица 2. Результаты сравнения алгоритмов  
Table 2. Algorithm comparison results

Критерий сравнения	Алгоритм					
	XУ-алгоритм	Алгоритм поворота модели	Маршрутизация с учетом перегрузок	Отказоустойчивая маршрутизация	Маршрутизация с учетом QoS	Алгоритм муравьиной колонии
Задержка и пропускная способность						
Адаптивность						
Отказоустойчивость						
Сложность						

На графике зависимости задержки от нагрузки на сеть (рис. 2, а) видно, что XУ-алгоритм показывает стабильное увеличение задержки с ростом нагрузки, но относительно низкое по сравнению с другими алгоритмами при низкой нагрузке. Алгоритм поворота модели демонстрирует улучшенную производительность по сравнению с XУ-алгоритмом, особенно при средних уровнях нагрузки. Маршрутизация с учетом перегрузок имеет минимальную задержку при низкой и средней нагрузках, но начинает увеличиваться при высокой нагрузке. Отказоустойчивая маршрутизация показывает изначально высокую задержку, что объясняется устойчивостью к сбоям. У маршрутизации с учетом QoS задержка растет с увеличением нагрузки, показывая хорошую производительность при средних уровнях нагрузки. Алгоритм муравьиной колонии демонстрирует минимальную задержку при низкой и средней нагрузках, но увеличивается при высокой нагрузке, отражая время на обучение и адаптацию маршрутов.

На графике пропускной способности от нагрузки на сеть (рис. 2, б) XУ-алгоритм показывает высокую пропускную способность, достигая почти максимальных

значений при высокой нагрузке. Алгоритм поворота модели близок по производительности к XУ-алгоритму, с небольшими улучшениями при высокой нагрузке. Маршрутизация с учетом перегрузок достигает наивысшей пропускной способности, особенно при средней и высокой нагрузках, эффективно распределяя трафик. Отказоустойчивая маршрутизация демонстрирует более низкую пропускную способность по сравнению с другими алгоритмами, что компенсируется отказоустойчивостью. Маршрутизация с учетом QoS показывает высокую пропускную способность, оптимизированную для приоритизации трафика. Алгоритм муравьиной колонии близок к максимальной пропускной способности при всех уровнях нагрузки, благодаря динамическому выбору маршрутов.

Полученные графики зависимостей (рис. 2) иллюстрируют то, как различные алгоритмы маршрутизации справляются с увеличением нагрузки на сеть, показывая их сильные и слабые стороны в условиях различных сценариев.

На следующем этапе работы предлагается разработка и верификация модели СенК. Основное внимание

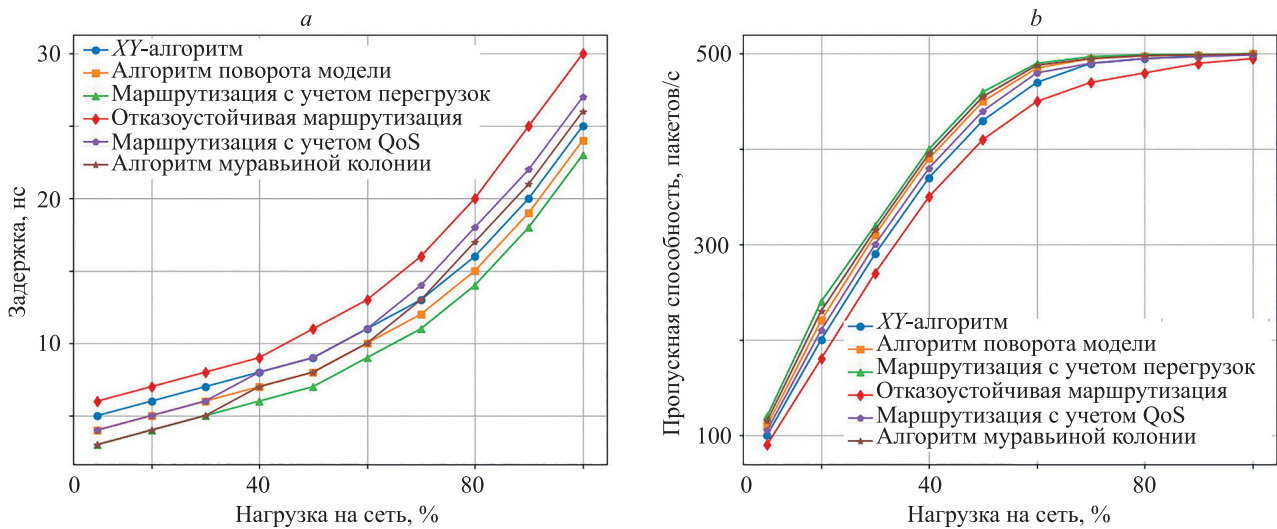


Рис. 2. Зависимости задержки (а) и пропускной способности (б) от нагрузки на сеть

Fig. 2. Graphs of latency and throughput vs. the network load



будет уделено созданию точной и масштабируемой модели, которая позволит эмулировать различные архитектурные и операционные сценарии в СенК.

Среди ожидаемых результатов выделим: настраиваемую модель СенК (способную адаптироваться к различным архитектурным требованиям); проведение «чистого» и полунатурного моделирования (с использованием программного обеспечения и программируемой логической интегральной схемы (ПЛИС) соответственно); разработанный набор тестов для валидации модели (включая проверку производительности, пропускной способности и задержек); сравнение полученной модели с существующими (что позволит оценить ее преимущества).

Необходимость экспериментальных исследований, в частности «чистого» моделирования, можно объяснить использованием программных средств для моделирования работы СенК без физического воплощения элементов. Данный метод позволит снизить затраты и время на эксперименты. Быстро модифицируя параметры системы, можно протестировать широкий спектр сценариев и архитектурных решений и оценить их влияние на общую производительность. Применение полунатурного моделирования, необходимо так как применение ПЛИС позволяет создавать более реалистичные модели работы СенК. Такой подход поможет оценить реальные физические ограничения и потенциальные неполадки, которые могут не быть очевидны при «чистом» моделировании.

Предложенное направление дальнейшей работы позволит не только подтвердить теоретические разработки на практике, но и значительно упростить процесс тестирования и внедрения архитектурных решений для СенК. Это также способствует более глубокому пониманию ограничений и возможностей СенК в условиях реальной эксплуатации.

Развитие новых стратегий маршрутизации, способных адаптироваться к изменяющимся условиям и требованиям, будет играть ключевую роль в обеспечении будущего развития интегральных схем и многопроцессорных систем на кристалле [46–48]. Это особенно важно для приложений, требующих высокой произво-

дительности, надежности и энергоэффективности, таких как большие многопроцессорные системы, интеллектуальные транспортные системы и распределенные вычислительные сети.

## Заключение

Выполнен анализ существующих алгоритмов маршрутизации в сетях на кристалле. Показано, что эффективность маршрутизации в сетях на кристалле варьируется в зависимости от выбора между основными критериями: производительностью, адаптивностью, отказоустойчивостью и сложностью реализации. Так, например, наиболее простые алгоритмы маршрутизации, хотя и не представляют сложности в реализации, ограничены в адаптивности и отказоустойчивости, в то время как более сложные алгоритмы, предлагают значительные преимущества в плане адаптивности и отказоустойчивости, но требуют более сложной реализации и управления. Определение оптимального алгоритма маршрутизации требует глубокого понимания как специфики архитектуры сетей на кристалле, так и характеристик трафика приложения. В будущих исследованиях следует сосредоточить усилия на разработке и оптимизации гибридных алгоритмов маршрутизации, которые могли бы сочетать в себе преимущества детерминированных и адаптивных методов, обеспечивая таким образом высокую производительность и отказоустойчивость при минимальной сложности реализации.

Проведенное исследование можно взять как основу для дальнейшей работы в этой области — разработки собственного алгоритма маршрутизации для сетей на кристалле, включающего в себя разработку прототипа на программируемой логической интегральной схеме. Кроме того, значительный интерес представляет разработка методик точного моделирования и оценки производительности различных алгоритмов в условиях динамически изменяющегося трафика и топологии сети, что позволит еще более эффективно подбирать алгоритмы маршрутизации под конкретные задачи и условия эксплуатации.

## Литература

1. Benini L., De Micheli G. Networks on chip: a new paradigm for systems on chip design // *Proc. of the Design, Automation and Test in Europe Conference and Exhibition*. 2002. P. 418–419. <https://doi.org/10.1109/DATE.2002.998307>
2. Bjerregaard T., Mahadevan S. A survey of research and practices of Network-on-chip // *ACM Computing Surveys*. 2006. V. 38. N 1. P. 1. <https://doi.org/10.1145/1132952.1132953>
3. Dielissen J., Goossens K., Rijpkema E. Concepts and Implementation of the Philips Network-on-Chip, 2003 [Электронный ресурс]. URL: <https://www.es.ele.tue.nl/~kgoossens/2003-ipsoc.pdf> (дата обращения: 20.02.2024).
4. Bertozzi D., Benini L., De Micheli G. Energy-efficient network-on-chip design // *Ultra Low-Power Electronics and Design*, 2004. P. 214–232. [https://doi.org/10.1007/1-4020-8076-X\\_12](https://doi.org/10.1007/1-4020-8076-X_12)
5. Mazumdar S., Scionti A., Portero A., Martinović J., Terzo O. A scalable and low-power FPGA-aware network-on-chip architecture // *Advances in Intelligent Systems and Computing*. 2018. V. 611. P. 407–420. [https://doi.org/10.1007/978-3-319-61566-0\\_37](https://doi.org/10.1007/978-3-319-61566-0_37)

## References

1. Benini L., De Micheli G. Networks on chip: a new paradigm for systems on chip design. *Proc. of the Design, Automation and Test in Europe Conference and Exhibition*, 2002, pp. 418–419. <https://doi.org/10.1109/DATE.2002.998307>
2. Bjerregaard T., Mahadevan S. A survey of research and practices of Network-on-chip. *ACM Computing Surveys*, 2006, vol. 38, no. 1, pp. 1. <https://doi.org/10.1145/1132952.1132953>
3. Dielissen J., Goossens K., Rijpkema E. *Concepts and Implementation of the Philips Network-on-Chip*, 2003. Available at: <https://www.es.ele.tue.nl/~kgoossens/2003-ipsoc.pdf> (accessed: 20.02.2024).
4. Bertozzi D., Benini L., De Micheli G. Energy-efficient network-on-chip design. *Ultra Low-Power Electronics and Design*, 2004, pp. 214–232. [https://doi.org/10.1007/1-4020-8076-X\\_12](https://doi.org/10.1007/1-4020-8076-X_12)
5. Mazumdar S., Scionti A., Portero A., Martinović J., Terzo O. A scalable and low-power FPGA-aware network-on-chip architecture. *Advances in Intelligent Systems and Computing*, 2018, vol. 611, pp. 407–420. [https://doi.org/10.1007/978-3-319-61566-0\\_37](https://doi.org/10.1007/978-3-319-61566-0_37)

6. Palesi M., Daneshtalab M. *Routing Algorithms in Networks-on-Chip*. Springer, 2014. 410 p. <https://doi.org/10.1007/978-1-4614-8274-1>
7. Khan K., Pasricha S. A reinforcement learning framework with region-awareness and shared path experience for efficient routing in networks-on-chip // *IEEE Design & Test*. 2023. V. 40. N 6. P. 76–85. <https://doi.org/10.1109/MDAT.2023.3306719>
8. Dally W.J., Towles B. Route packets, not wires: On-chip interconnection networks // *Proc. of the 38<sup>th</sup> Design Automation Conference*. 2001. P. 684–689. <https://doi.org/10.1109/dac.2001.935594>
9. Murali S., De Micheli G. Bandwidth-constrained mapping of cores onto NoC architectures // *Proceedings Design, Automation and Test in Europe Conference and Exhibition*. V. 2. 2004. P. 896–901. <https://doi.org/10.1109/DATE.2004.1269002>
10. Ma S., Wang Z., Enright Jerger N., Shen L., Xiao N. Novel flow control for fully adaptive routing in cache-coherent NoCs // *IEEE Transactions on Parallel and Distributed Systems*. 2014. V. 25. N 9. P. 2397–2407. <https://doi.org/10.1109/TPDS.2013.166>
11. Nicopoulos C.A., Park D., Kim J., Vijaykrishnan N., Yousif M.S., Das C.R. ViChaR: A dynamic virtual channel regulator for network-on-chip routers // *Proc. of the 39<sup>th</sup> Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'06)*. 2006. P. 333–346. <https://doi.org/10.1109/MICRO.2006.50>
12. Owens J.D., Dally W.J., Ho R., Jayasimha D.N., Keckler S.W., Peh L.-S. Research challenges for on-chip interconnection networks // *IEEE Micro*. 2007. V. 27. N 5. P. 96–108. <https://doi.org/10.1109/MM.2007.4378787>
13. Mak T., Cheung P.Y.K., Lam K.-P., Luk W. Adaptive routing in network-on-chips using a dynamic-programming network // *IEEE Transactions on Industrial Electronics*. 2011. V. 58. N 8. P. 3701–3716. <https://doi.org/10.1109/TIE.2010.2081953>
14. Paliwal K.K., Gaur M.S., Laxmi V., Janyani V. Performance analysis of guaranteed throughput and best effort traffic in network-on-chip under different traffic scenario // *Proc. of the International Conference on Future Networks*. 2009. P. 74–78. <https://doi.org/10.1109/ICFN.2009.45>
15. Tedesco L., Mello A., Garibotti D., Calazans N., Moraes F. Traffic generation and performance evaluation for mesh-based NoCs // *Proc. of the 18<sup>th</sup> Symposium on Integrated Circuits and Systems Design*. 2005. P. 184–189. <https://doi.org/10.1109/SBCCI.2005.4286854>
16. Soteriou V., Wang H., Peh L. A statistical traffic model for on-chip interconnection networks // *Proc. of the 14<sup>th</sup> IEEE International Symposium on Modeling, Analysis, and Simulation*. 2006. P. 104–116. <https://doi.org/10.1109/MASCOTS.2006.9>
17. Velayudham S., Rajagopal S., Kathirvel S., Alhadidi B. An overview of multicast routing algorithms in network on chip // *Learning and Analytics in Intelligent Systems*. 2021. V. 21. P. 163–178. [https://doi.org/10.1007/978-3-030-65407-8\\_14](https://doi.org/10.1007/978-3-030-65407-8_14)
18. Xiang D., Yu Z., Wu J. Deadlock-free fully adaptive routing in irregular networks without virtual channels // *Proc. of the 12<sup>th</sup> IEEE International Conference on Trust, Security and Privacy in Computing and Communications*. 2013. P. 983–990. <https://doi.org/10.1109/TrustCom.2013.120>
19. Palesi M., Holmsmark R., Kumar S., Catania V. A methodology for design of application specific deadlock-free routing algorithms for NoC systems // *Proc. of the 4<sup>th</sup> International Conference on Hardware/Software Codesign and System Synthesis*. 2006. P. 142–147. <https://doi.org/10.1145/1176254.1176289>
20. Taheri E., Pasricha S., Nikdast M. DeFT: A Deadlock-free and fault-tolerant routing algorithm for 2.5D chiplet networks // *Proc. of the Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 2022. P. 1047–1052. <https://doi.org/10.23919/DATE54114.2022.9774617>
21. Valinataj M., Mohammadi S., Plosila J., Liljeberg P. A fault-tolerant and congestion-aware routing algorithm for Networks-on-Chip // *Proc. of the 13<sup>th</sup> IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems*. 2010. P. 139–144. <https://doi.org/10.1109/DDECS.2010.5491798>
22. Patooghy A., Tabkhi H., Miremadi S.G. An efficient method to reliable data transmission in Network-on-Chips // *Proc. of the 13<sup>th</sup> Euromicro Conference on Digital System Design: Architectures, Methods and Tools*. 2010. P. 467–474. <https://doi.org/10.1109/DSD.2010.23>
23. Lysne O., Montanana J.M., Flich J., Duato J., Pinkston T.M., Skei T. An efficient and deadlock-free network reconfiguration protocol // *IEEE Transactions on Computers*. 2008. V. 57. N 6. P. 762–779. <https://doi.org/10.1109/TC.2008.31>
6. Palesi M., Daneshtalab M. *Routing Algorithms in Networks-on-Chip*. Springer, 2014. 410 p. <https://doi.org/10.1007/978-1-4614-8274-1>
7. Khan K., Pasricha S. A reinforcement learning framework with region-awareness and shared path experience for efficient routing in networks-on-chip. *IEEE Design & Test*, 2023, vol. 40, no. 6, pp. 76–85. <https://doi.org/10.1109/MDAT.2023.3306719>
8. Dally W.J., Towles B. Route packets, not wires: On-chip interconnection networks. *Proc. of the 38<sup>th</sup> Design Automation Conference*, 2001, pp. 684–689. <https://doi.org/10.1109/dac.2001.935594>
9. Murali S., De Micheli G. Bandwidth-constrained mapping of cores onto NoC architectures. *Proceedings Design, Automation and Test in Europe Conference and Exhibition*. V. 2, 2004, pp. 896–901. <https://doi.org/10.1109/DATE.2004.1269002>
10. Ma S., Wang Z., Enright Jerger N., Shen L., Xiao N. Novel flow control for fully adaptive routing in cache-coherent NoCs. *IEEE Transactions on Parallel and Distributed Systems*, 2014, vol. 25, no. 9, pp. 2397–2407. <https://doi.org/10.1109/TPDS.2013.166>
11. Nicopoulos C.A., Park D., Kim J., Vijaykrishnan N., Yousif M.S., Das C.R. ViChaR: A dynamic virtual channel regulator for network-on-chip routers. *Proc. of the 39<sup>th</sup> Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'06)*, 2006, pp. 333–346. <https://doi.org/10.1109/MICRO.2006.50>
12. Owens J.D., Dally W.J., Ho R., Jayasimha D.N., Keckler S.W., Peh L.-S. Research challenges for on-chip interconnection networks. *IEEE Micro*, 2007, vol. 27, no. 5, pp. 96–108. <https://doi.org/10.1109/MM.2007.4378787>
13. Mak T., Cheung P.Y.K., Lam K.-P., Luk W. Adaptive routing in network-on-chips using a dynamic-programming network. *IEEE Transactions on Industrial Electronics*, 2011, vol. 58, no. 8, pp. 3701–3716. <https://doi.org/10.1109/TIE.2010.2081953>
14. Paliwal K.K., Gaur M.S., Laxmi V., Janyani V. Performance analysis of guaranteed throughput and best effort traffic in network-on-chip under different traffic scenario. *Proc. of the International Conference on Future Networks*, 2009, pp. 74–78. <https://doi.org/10.1109/ICFN.2009.45>
15. Tedesco L., Mello A., Garibotti D., Calazans N., Moraes F. Traffic generation and performance evaluation for mesh-based NoCs. *Proc. of the 18<sup>th</sup> Symposium on Integrated Circuits and Systems Design*, 2005, pp. 184–189. <https://doi.org/10.1109/SBCCI.2005.4286854>
16. Soteriou V., Wang H., Peh L. A statistical traffic model for on-chip interconnection networks. *Proc. of the 14<sup>th</sup> IEEE International Symposium on Modeling, Analysis, and Simulation*, 2006, pp. 104–116. <https://doi.org/10.1109/MASCOTS.2006.9>
17. Velayudham S., Rajagopal S., Kathirvel S., Alhadidi B. An overview of multicast routing algorithms in network on chip. *Learning and Analytics in Intelligent Systems*, 2021, vol. 21, pp. 163–178. [https://doi.org/10.1007/978-3-030-65407-8\\_14](https://doi.org/10.1007/978-3-030-65407-8_14)
18. Xiang D., Yu Z., Wu J. Deadlock-free fully adaptive routing in irregular networks without virtual channels. *Proc. of the 12<sup>th</sup> IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2013, pp. 983–990. <https://doi.org/10.1109/TrustCom.2013.120>
19. Palesi M., Holmsmark R., Kumar S., Catania V. A methodology for design of application specific deadlock-free routing algorithms for NoC systems. *Proc. of the 4<sup>th</sup> International Conference on Hardware/Software Codesign and System Synthesis*, 2006, pp. 142–147. <https://doi.org/10.1145/1176254.1176289>
20. Taheri E., Pasricha S., Nikdast M. DeFT: A Deadlock-free and fault-tolerant routing algorithm for 2.5D chiplet networks. *Proc. of the Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2022, pp. 1047–1052. <https://doi.org/10.23919/DATE54114.2022.9774617>
21. Valinataj M., Mohammadi S., Plosila J., Liljeberg P. A fault-tolerant and congestion-aware routing algorithm for Networks-on-Chip. *Proc. of the 13<sup>th</sup> IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems*, 2010, pp. 139–144. <https://doi.org/10.1109/DDECS.2010.5491798>
22. Patooghy A., Tabkhi H., Miremadi S.G. An efficient method to reliable data transmission in Network-on-Chips. *Proc. of the 13<sup>th</sup> Euromicro Conference on Digital System Design: Architectures, Methods and Tools*, 2010, pp. 467–474. <https://doi.org/10.1109/DSD.2010.23>
23. Lysne O., Montanana J.M., Flich J., Duato J., Pinkston T.M., Skei T. An efficient and deadlock-free network reconfiguration protocol. *IEEE Transactions on Computers*, 2008, vol. 57, no. 6, pp. 762–779. <https://doi.org/10.1109/TC.2008.31>

24. Adamu G., Chejara P., Garko A. Review of deterministic routing algorithm for Network-on-Chip // *International Journal of Advance Research in Science and Engineering*. 2015. V. 4. Spec. Issue 1. P. 123–128.
25. Mahendra C., Gaikwad M., Patrikar R. Review of XY routing algorithm for network-on-chip architecture // *International Journal of Computer and Communication Technology*. 2016. V. 43. P. 20–23. <https://doi.org/10.47893/IJCCT.2016.1384>
26. Kumar M., Laxmi V., Gaur M.S., Daneshtalab M., Zwolinski M. A novel non-minimal turn model for highly adaptive routing in 2D NoCs // *Proc. of the 22<sup>nd</sup> International Conference on Very Large Scale Integration (VLSI-SoC)*. 2014. P. 1–6. <https://doi.org/10.1109/VLSI-SoC.2014.7004192>
27. Brown J.W. *Adaptive Network on Chip Routing using the Turn Model: Master's Theses and Capstones*. 2013 [Электронный ресурс]. URL: <https://scholars.unh.edu/thesis/779> (дата обращения: 20.02.2024)
28. Ahmad K., Sethi M.A.J., Ullah R., Ahmed I., Ullah A., Jan N., Karami G.M. Congestion-aware routing algorithm for NoC using data packets // *Wireless Communications and Mobile Computing*. 2021. V. 2921. <https://doi.org/10.1155/2021/8588646>
29. Ponnann S., Kumar T.A., Hemakumar V.S., Natarajan S., Shah M.A. Congestion aware low power on chip protocols with network on chip with cloud security // *Journal of Cloud Computing*. 2022. V. 11. N 1. P 41. <https://doi.org/10.1186/s13677-022-00307-4>
30. Akbar R., Etedalpour A.A., Safaei F. An efficient fault-tolerant routing algorithm in NoCs to tolerate permanent faults // *The Journal of Supercomputing*. 2016. V. 72. N 12. P. 4629–4650. <https://doi.org/10.1007/s11227-016-1749-0>
31. Zhang Z., Serwe W., Wu J., Yoneda T., Zheng H., Myers C. Formal analysis of a fault-tolerant routing algorithm for a Network-on-Chip // *Lecture Notes in Computer Science*. 2014. V. 8718. P. 48–62. [https://doi.org/10.1007/978-3-319-10702-8\\_4](https://doi.org/10.1007/978-3-319-10702-8_4)
32. Carara E., Calazans N., Moraes F. Managing QoS flows at task level in NoC-based MPSoCs // *Proc. of the 17<sup>th</sup> IFIP International Conference on Very Large Scale Integration. (VLSI-SoC)*. 2009. P. 133–138. <https://doi.org/10.1109/VLSISOC.2009.6041343>
33. Xie P., Gu H. Intelligent bees for QoS routing in Networks-on-Chip // *Proc. of the Second Pacific-Asia Conference on Circuits, Communications and System*. 2010. P. 311–314. <https://doi.org/10.1109/PACCS.2010.5626974>
34. Nedjah N., de Macedo Mourelle L. Routing in Network-on-Chips using ant colony optimization // *Studies in Computational Intelligence*. 2014. V. 529. P. 173–198. [https://doi.org/10.1007/978-3-319-03110-1\\_11](https://doi.org/10.1007/978-3-319-03110-1_11)
35. Nedjah N., Silva Junior L., De Macedo Mourelle L. Congestion-aware an colony based routing algorithms for efficient application execution on Network-on-Chip platform // *Expert Systems with Applications*. 2013. V. 40. N 16. P. 6661–6673. <https://doi.org/10.1016/j.eswa.2013.06.005>
36. Ahmad K., Sethi M.A.J. Review of Network on Chip routing algorithms // *EAI Endorsed Transactions on Context-aware Systems and Applications*. 2020. V. 7. N 2. P. e5. <https://doi.org/10.4108/eai.23-12-2020.167793>
37. Xu Y., Zhou J., Liu S. Research and analysis of routing algorithms for NoC // *Proc. of the 3<sup>rd</sup> International Conference on Computer Research and Development*. 2011. P. 98–102. <https://doi.org/10.1109/ICCRD.2011.5764092>
38. Uma R., Sarojadevi H., Sanju V. Network-On-Chip (NoC) - routing techniques: a study and analysis // *Proc. of the Global Conference for Advancement in Technology (GCAT)*. 2019. P. 1–6. <https://doi.org/10.1109/GCAT47503.2019.8978403>
39. Sharifi Z., Mohammadi S., Sirjani M. Comparison of NoC Routing Algorithms Using Formal Methods. 2013 [Электронный ресурс]. URL: <https://rebeca-lang.org/assets/papers/2013/NoC-Routing-Algorithms-Comparison.pdf> (дата обращения: 20.02.2024).
40. Saliu M., Momoh M., Chinedu P., Nwankwo W., Daniel A. Comparative performance analysis of selected routing algorithms by load variation of 2-dimensional mesh topology based Network-On-Chip // *ELEKTRIKA-Journal of Electrical Engineering*. 2021. V. 20. N 3. P. 1–6. <https://doi.org/10.11113/elektrika.v20n3.249>
41. Kaleem M., Isnin I.F.B. A survey on network on chip routing algorithms criteria // *Advances in Intelligent Systems and Computing*. 2021. V. 1188. P. 455–466. [https://doi.org/10.1007/978-981-15-6048-4\\_40](https://doi.org/10.1007/978-981-15-6048-4_40)
42. Singh J.K., Swain A.K., Kamal Reddy T.N., Mahapatra K.K. Performance evaluation of different routing algorithms in Network
24. Adamu G., Chejara P., Garko A. Review of deterministic routing algorithm for Network-on-Chip. *International Journal of Advance Research in Science and Engineering*, 2015, vol. 4, spec. issue 1, pp. 123–128.
25. Mahendra C., Gaikwad M., Patrikar R. Review of XY routing algorithm for network-on-chip architecture. *International Journal of Computer and Communication Technology*, 2016, vol. 43, pp. 20–23. <https://doi.org/10.47893/IJCCT.2016.1384>
26. Kumar M., Laxmi V., Gaur M.S., Daneshtalab M., Zwolinski M. A novel non-minimal turn model for highly adaptive routing in 2D NoCs. *Proc. of the 22<sup>nd</sup> International Conference on Very Large Scale Integration (VLSI-SoC)*, 2014, pp. 1–6. <https://doi.org/10.1109/VLSI-SoC.2014.7004192>
27. Brown J.W. *Adaptive Network on Chip Routing using the Turn Model: Master's Theses and Capstones*. 2013. Available at: <https://scholars.unh.edu/thesis/779> (accessed: 20.02.2024)
28. Ahmad K., Sethi M.A.J., Ullah R., Ahmed I., Ullah A., Jan N., Karami G.M. Congestion-aware routing algorithm for NoC using data packets. *Wireless Communications and Mobile Computing*, 2021, vol. 2921. <https://doi.org/10.1155/2021/8588646>
29. Ponnann S., Kumar T.A., Hemakumar V.S., Natarajan S., Shah M.A. Congestion aware low power on chip protocols with network on chip with cloud security. *Journal of Cloud Computing*, 2022, vol. 11, no. 1, pp 41. <https://doi.org/10.1186/s13677-022-00307-4>
30. Akbar R., Etedalpour A.A., Safaei F. An efficient fault-tolerant routing algorithm in NoCs to tolerate permanent faults. *The Journal of Supercomputing*, 2016, vol. 72, no. 12, pp. 4629–4650. <https://doi.org/10.1007/s11227-016-1749-0>
31. Zhang Z., Serwe W., Wu J., Yoneda T., Zheng H., Myers C. Formal analysis of a fault-tolerant routing algorithm for a Network-on-Chip. *Lecture Notes in Computer Science*, 2014, vol. 8718, pp. 48–62. [https://doi.org/10.1007/978-3-319-10702-8\\_4](https://doi.org/10.1007/978-3-319-10702-8_4)
32. Carara E., Calazans N., Moraes F. Managing QoS flows at task level in NoC-based MPSoCs. *Proc. of the 17<sup>th</sup> IFIP International Conference on Very Large Scale Integration. (VLSI-SoC)*, 2009, pp. 133–138. <https://doi.org/10.1109/VLSISOC.2009.6041343>
33. Xie P., Gu H. Intelligent bees for QoS routing in Networks-on-Chip. *Proc. of the Second Pacific-Asia Conference on Circuits, Communications and System*, 2010, pp. 311–314. <https://doi.org/10.1109/PACCS.2010.5626974>
34. Nedjah N., de Macedo Mourelle L. Routing in Network-on-Chips using ant colony optimization. *Studies in Computational Intelligence*, 2014, vol. 529, pp. 173–198. [https://doi.org/10.1007/978-3-319-03110-1\\_11](https://doi.org/10.1007/978-3-319-03110-1_11)
35. Nedjah N., Silva Junior L., De Macedo Mourelle L. Congestion-aware an colony based routing algorithms for efficient application execution on Network-on-Chip platform. *Expert Systems with Applications*, 2013, vol. 40, no. 16, pp. 6661–6673. <https://doi.org/10.1016/j.eswa.2013.06.005>
36. Ahmad K., Sethi M.A.J. Review of Network on Chip routing algorithms. *EAI Endorsed Transactions on Context-aware Systems and Applications*, 2020, vol. 7, no. 2, pp. e5. <https://doi.org/10.4108/eai.23-12-2020.167793>
37. Xu Y., Zhou J., Liu S. Research and analysis of routing algorithms for NoC. *Proc. of the 3<sup>rd</sup> International Conference on Computer Research and Development*, 2011, pp. 98–102. <https://doi.org/10.1109/ICCRD.2011.5764092>
38. Uma R., Sarojadevi H., Sanju V. Network-On-Chip (NoC) - routing techniques: a study and analysis. *Proc. of the Global Conference for Advancement in Technology (GCAT)*, 2019, pp. 1–6. <https://doi.org/10.1109/GCAT47503.2019.8978403>
39. Sharifi Z., Mohammadi S., Sirjani M. *Comparison of NoC Routing Algorithms Using Formal Methods*. 2013. Available at: <https://rebeca-lang.org/assets/papers/2013/NoC-Routing-Algorithms-Comparison.pdf> (accessed: 20.02.2024).
40. Saliu M., Momoh M., Chinedu P., Nwankwo W., Daniel A. Comparative performance analysis of selected routing algorithms by load variation of 2-dimensional mesh topology based Network-On-Chip. *ELEKTRIKA-Journal of Electrical Engineering*, 2021, vol. 20, no. 3, pp. 1–6. <https://doi.org/10.11113/elektrika.v20n3.249>
41. Kaleem M., Isnin I.F.B. A survey on network on chip routing algorithms criteria. *Advances in Intelligent Systems and Computing*, 2021, vol. 1188, pp. 455–466. [https://doi.org/10.1007/978-981-15-6048-4\\_40](https://doi.org/10.1007/978-981-15-6048-4_40)
42. Singh J.K., Swain A.K., Kamal Reddy T.N., Mahapatra K.K. Performance evaluation of different routing algorithms in Network

- on Chip // Proc. of the IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), 2013, P. 180–185. <https://doi.org/10.1109/PrimeAsia.2013.6731201>
43. Soni N., Deshmukh K. Comparison between three different types of routing algorithms of Network on Chip // Springer Proceedings in Physics, 2015, V. 166, P. 447–459. [https://doi.org/10.1007/978-81-322-2367-2\\_56](https://doi.org/10.1007/978-81-322-2367-2_56)
44. Palesi M., Holsmark R., Kumar S., Catania V. Application-specific routing algorithms for low power network on chip design // Low Power Networks-on-Chip. Springer, 2011, P. 113–150. [https://doi.org/10.1007/978-1-4419-6911-8\\_5](https://doi.org/10.1007/978-1-4419-6911-8_5)
45. Zulkefli F.W., Ehkan P., Warip M.N.M., Phing N.Y., Zakaria F.F. A comparative review of adaptive routing approach for Network-on-Chip router architecture // Lecture Notes on Data Engineering and Communications Technologies, 2018, V. 5, P. 247–254. [https://doi.org/10.1007/978-3-319-59427-9\\_27](https://doi.org/10.1007/978-3-319-59427-9_27)
46. Hesham S., Rettkowski J., Göhringer D., Abd El Ghany M.A. Survey on real-time network-on-chip architectures // Lecture Notes in Computer Science, 2015, V. 9040, P. 191–202. [https://doi.org/10.1007/978-3-319-16214-0\\_16](https://doi.org/10.1007/978-3-319-16214-0_16)
47. Venkataraman N.L., Kumar R. Design and analysis of application specific network on chip for reliable custom topology // Computer Networks, 2019, V. 158, P. 69–76. <https://doi.org/10.1016/j.comnet.2019.03.014>
48. Madalozzo G., Indrusiak L.S., Moraes F.G. Mapping of real-time applications on a packet switching NoC-based MPSoC // Proc. of the IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2016, P. 640–643. <https://doi.org/10.1109/ICECS.2016.7841283>
- on Chip. Proc. of the IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), 2013, pp. 180–185. <https://doi.org/10.1109/PrimeAsia.2013.6731201>
43. Soni N., Deshmukh K. Comparison between three different types of routing algorithms of Network on Chip. Springer Proceedings in Physics, 2015, vol. 166, pp. 447–459. [https://doi.org/10.1007/978-81-322-2367-2\\_56](https://doi.org/10.1007/978-81-322-2367-2_56)
44. Palesi M., Holsmark R., Kumar S., Catania V. Application-specific routing algorithms for low power network on chip design. Low Power Networks-on-Chip. Springer, 2011, pp. 113–150. [https://doi.org/10.1007/978-1-4419-6911-8\\_5](https://doi.org/10.1007/978-1-4419-6911-8_5)
45. Zulkefli F.W., Ehkan P., Warip M.N.M., Phing N.Y., Zakaria F.F. A comparative review of adaptive routing approach for Network-on-Chip router architecture. Lecture Notes on Data Engineering and Communications Technologies, 2018, vol. 5, pp. 247–254. [https://doi.org/10.1007/978-3-319-59427-9\\_27](https://doi.org/10.1007/978-3-319-59427-9_27)
46. Hesham S., Rettkowski J., Göhringer D., Abd El Ghany M.A. Survey on real-time network-on-chip architectures. Lecture Notes in Computer Science, 2015, vol. 9040, pp. 191–202. [https://doi.org/10.1007/978-3-319-16214-0\\_16](https://doi.org/10.1007/978-3-319-16214-0_16)
47. Venkataraman N.L., Kumar R. Design and analysis of application specific network on chip for reliable custom topology. Computer Networks, 2019, vol. 158, pp. 69–76. <https://doi.org/10.1016/j.comnet.2019.03.014>
48. Madalozzo G., Indrusiak L.S., Moraes F.G. Mapping of real-time applications on a packet switching NoC-based MPSoC. Proc. of the IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2016, pp. 640–643. <https://doi.org/10.1109/ICECS.2016.7841283>

#### Авторы

**Бондаренко Михаил Игоревич** — аспирант, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация; инженер, ООО «Синтакор, Санкт-Петербург, 196247, Российская Федерация, <https://orcid.org/0009-0002-6816-4055>, [bondarenko-m-i@yandex.ru](mailto:bondarenko-m-i@yandex.ru)  
**Платунов Алексей Евгеньевич** — доктор технических наук, профессор, профессор, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, [sc 35318291200](https://orcid.org/0000-0003-3003-3949), <https://orcid.org/0000-0003-3003-3949>, [aeplatunov@itmo.ru](mailto:aeplatunov@itmo.ru)

Статья поступила в редакцию 11.04.2024  
 Одобрена после рецензирования 18.07.2024  
 Принята к печати 26.09.2024

#### Authors

**Mikhail I. Bondarenko** — PhD Student, ITMO University, Saint Petersburg, 197101, Russian Federation; Hardware Engineer, Syntacore, Saint Petersburg, 196247, Russian Federation, <https://orcid.org/0009-0002-6816-4055>, [bondarenko-m-i@yandex.ru](mailto:bondarenko-m-i@yandex.ru)  
**Alexey E. Platunov** — D.Sc., Full Professor, ITMO University, Saint Petersburg, 197101, Russian Federation, [sc 35318291200](https://orcid.org/0000-0003-3003-3949), <https://orcid.org/0000-0003-3003-3949>, [aeplatunov@itmo.ru](mailto:aeplatunov@itmo.ru)

Received 11.04.2024  
 Approved after reviewing 18.07.2024  
 Accepted 26.09.2024



Работа доступна по лицензии  
 Creative Commons  
 «Attribution-NonCommercial»