

УДК 004.3

Моделирование маршрутизаторов на многоядерных сетевых процессорах

Грищенко В.И., Ладыженский Ю.В.

Донецкий национальный технический университет
victor.grischenko@gmail.com, ly@cs.dgtu.donetsk.ua

Abstract

Grischenko V., Ladyzhenskiy Y. Modeling of multi-core network processors based routers. This article describes analytical model for network multi-core processors. It based on queuing theory and takes in consideration pipeline and parallel packets processing. All processing cores have individual separated instruction and data caches and share single RAM interface.

Введение

В настоящее время требования к пропускной способности маршрутизаторов значительно превосходят возможности современных процессоров общего назначения [1]. Но разнобразные задачи, которые могут решаться маршрутизатором не позволяют использовать узкоспециализированные аппаратные решения [2]. Компромиссом могут выступить программируемые устройства со структурой, ориентированной на особенности обработки данных в компьютерных сетях. Такими устройствами стали сетевые процессоры [2].

Сетевой процессор (СП) — это специализированное устройство, которое решает основные, наиболее трудоемкие, задачи обработки пакетов в маршрутизаторе. Для увеличения пропускной способности сетевые процессоры часто используют многоядерную структуру и задача поиска наиболее производительной структуры СП является актуальной научной проблемой.

В настоящее время существует несколько направлений в моделировании сетевых процессоров [3]. В работах [4, 5] предлагается аналитическая модель, основанная на кривых поступления и кривых обслуживания. Кривая поступления описывает входной поток, который, проходя через узлы процессора, испытывает обслуживание и уменьшается на величину полученного обслуживания. Предложенный подход позволяет гибко варьировать структуру СП и характеристики входного потока, но сложности определения характеристик приложений и узлов делает его использование затруднительным.

Решение проблемы получения характеристик приложений было предложено в работах [6, 7]. В них был представлен механизм извлечения информации о характеристиках приложения из имитационного моделирования. И единожды полученные характеристики могут многократно использоваться при аналитическом моделировании различных структур СП. Но в используемой аналитической модели отсутствует возможность моделирования конвейерных структур, что

существенно снижает область их применения.

В работе [8] предложена аналитическая модель многоядерного сетевого процессора, основанная на теории массового обслуживания. В предложенной модели одно ядро выделено в качестве управляющего, которое подключает остальные вычислительные ядра по мере необходимости. Кроме того, модель не позволяет исследовать конвейерные структуры.

В статье авторами предлагается аналитическая модель для определения производительности сетевых процессоров на основе характеристик алгоритмов обработки пакетов. Характеристики алгоритмов могут получаться из статистики работы реальных приложений на реальной нагрузке аналогично методам, предложенным в [6, 7]. Модель позволяет исследовать конвейерные, параллельные и смешанные структуры СП.

Модель сетевого процессора

Обработка сетевых пакетов на маршрутизаторе обладает особенностями, использование которых может увеличить эффективность многоядерных СП: независимость обработки разных пакетов и линейность этапов обработки отдельного пакета [2].

Независимость обработки разных пакетов означает, что можно параллельно обрабатывать несколько пакетов без накладных расходов на синхронизацию между программными потоками. Линейность этапов обработки пакета позволяет обрабатывать пакеты в режиме конвейера. В процессе конвейерной обработки пакета возврат на предыдущие этапы обработки не требуется [1]. Это означает, что после заполнения конвейера, многопроцессорная система будет работать с максимальной производительностью.

Различные этапы обработки пакетов могут иметь различную трудоемкость распределение вычислительных ядер СП между этапами конвейера является важной задачей.

Структура многоядерного СП, которая позволяет использовать оба метода распараллеливания обработки сетевого потока данных, представлена на рисунке 1. Каждое вычислительное ядро состоит из арифметико-

логического устройства (АЛУ) и имеет отдельные кэши данных (K_d) и команд (K_k). Ядра разделены на несколько этапов конвейера и разделяют общий интерфейс к оперативной памяти (ОЗУ).

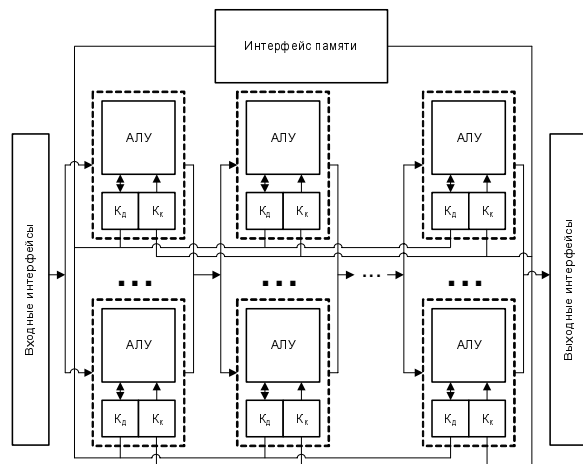


Рисунок 1 – Структура сетевого процессора

После прохождения входных сетевых интерфейсов пакет начинает обработку на первом этапе конвейера. Если этап содержит несколько вычислительных ядер, то пакет занимает любое свободное ядро. Если свободных ядер нет, то пакет ожидает завершения обработки предыдущих пакетов. После обработки на первом этапе пакет последовательно проходит остальные этапы и покидает СП через выходные сетевые интерфейсы. В случае возникновения ошибки (неправильная контрольная сумма, истекшее значение TTL и др.) пакет уничтожается [9].

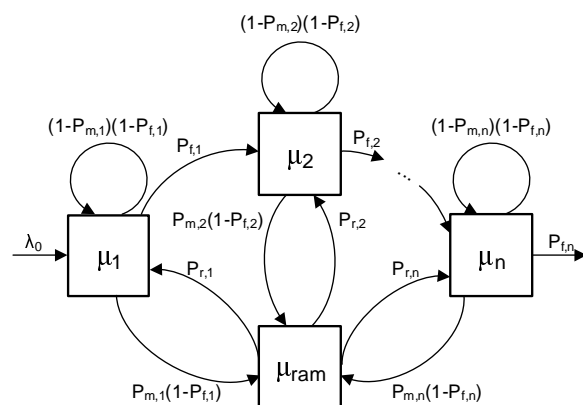


Рисунок 2 – Сеть СМО, моделирующая сетевой процессор

Многоядерный сетевой процессор можно представить сетью массового обслуживания (рисунок 2). Узлы $\mu_1 - \mu_n$ моделируют работу этапов конвейера обработки пакетов. Каждый узел содержит обслуживающие устройства, моделирующие вычислительные ядра, выделенные для выполнения соответствующего этапа конвейера.

Узел μ_{ram} моделирует работу блока ОЗУ. Он

имеет одно обслуживающее устройство, его интенсивность обработки зависит от характеристик используемого оборудования.

При поступлении пакета на вычислительное ядро выполняется одна инструкция из алгоритма его обработки, после чего, с вероятностью $P_{f,i}$ ($i = \overline{1, n}$) обработка может завершиться. Вероятность $P_{f,i}$ зависит от среднего числа инструкций ($N_{ins,i}$), необходимых для обработки пакета на i -м этапе конвейера:

$$P_{f,i} = \frac{1}{N_{ins,i} + 1} \quad (1)$$

В случае, если обработка не завершилась, то с вероятностью $P_{m,i}$ ($i = \overline{1, n}$) потребуется обращение к оперативной памяти за данными или кодом. Вероятность $P_{m,i}$ зависит от интенсивности обращений к ОЗУ и вероятностей промаха кэшей:

$$P_{m,i} = P_{dc,i}P_{d,i} + P_{ic,i} \quad (2)$$

где, $P_{d,i}$ — вероятность обращения за данными, во время обработки пакета на i -м этапе конвейера, $P_{dc,i}$ — вероятность промаха кэша данных, $P_{ic,i}$ — вероятность промаха кэша команд.

Вероятность $P_{d,i}$ может быть рассчитана как отношение обращений к ОЗУ за время обработки одного пакета ($N_{d,i}$) к общему количеству инструкций, выполненных за время его обработки ($N_{prc,i}$):

$$P_{d,i} = \frac{N_{d,i}}{N_{prc,i}} \quad (3)$$

После завершения обращения к ОЗУ обработка пакета продолжается на том же этапе конвейера, на котором она была прервана для обращения к памяти. Все этапы конвейера используют один модуль ОЗУ. Необходимо определить, какая часть выходного потока ОЗУ адресуется каждому отдельному этапу конвейера. Поскольку входной поток от i -го этапа конвейера к ОЗУ равен $P_{m,i}(1-P_{f,i})$, то вероятность перехода пакета из ОЗУ на i -й этап конвейера ($P_{r,i}$) определяется по формуле:

$$P_{r,i} = \frac{P_{m,i}(1-P_{f,i})}{\sum_{k=1}^n P_{m,k}(1-P_{f,k})} \quad (4)$$

Если пакет не завершил обработку, и не было обращения к оперативной памяти то, с вероятностью $(1-P_{m,i})(1-P_{f,i})$, он вновь поступает в очередь обработки этого же этапа конвейера.

По завершению обработки с вероятностью $P_{f,i}$ ($i = \overline{1, n-1}$) пакет передается следующему этапу конвейера. Выходной поток $P_{f,n}$ моделирует исходящий поток обработанных пакетов.

Пусть входной поток пакетов для сети СМО, представленной на рисунке 2, распределен по закону Пуассона с интенсивностью λ_0 . Тогда можно записать систему уравнений:

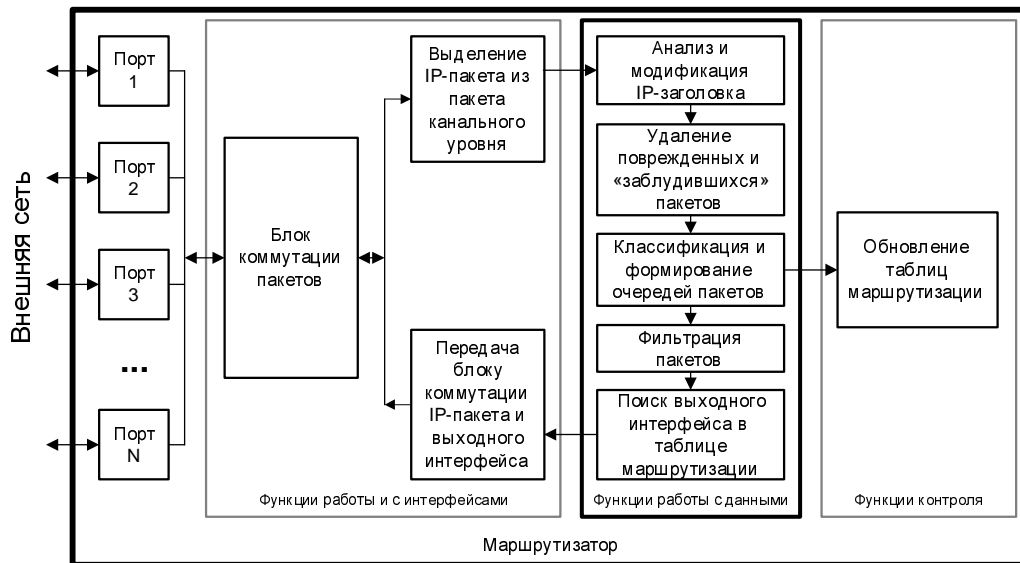


Рисунок 3 – Функциональная структура маршрутизатора

$$\begin{cases} \lambda_0 = \lambda_n P_{f_n} \\ \lambda_i = \lambda_{i-1} P_{f,i-1} + \lambda_i (1 - P_{m,i})(1 - P_{f,i}) + \lambda_{ram} P_{r,i}, i = \overline{1, n} \\ \lambda_{ram} = \sum_{i=1}^n \lambda_i P_{m,i} (1 - P_{f,i}) \end{cases} \quad (5)$$

где уравнения для λ_i ($i = \overline{1, n}$) соответствуют этапам конвейера, а λ_{ram} описывает входной поток блока ОЗУ.

Определим интенсивности обслуживания для всех узлов системы. Будем предполагать, что все вычислительные ядра одинаковы и могут обслуживать только одну инструкцию за такт. Тогда интенсивность обслуживания μ_i равна тактовой частоте процессора, выраженной в герцах (f_{pr}):

$$\mu_i = f_{pr} \quad (6)$$

Допустим, что блок ОЗУ не может обслуживать несколько запросов одновременно, тогда его интенсивность обслуживания будет равна:

$$\mu_{ram} = \frac{1}{t_{ram}} \quad (7)$$

где, t_{ram} - среднее время выполнения операции чтения/записи для ОЗУ.

Реализация функций маршрутизатора на сетевом процессоре

Маршрутизация сетевых пакетов — задача, которая выполняется на маршрутизаторах всех уровней. Под маршрутизацией понимается определение сетевого интерфейса адреса следующего маршрутизатора, через который пакет будет отправлен дальше к своему узлу назначения. Эта операция относится к сетевому уровню модели OSI [10].

В общем случае алгоритм маршрутизации состоит из трех этапов: выделение из заголовков

пакета адреса назначения; поиск в таблице маршрутизации выходного интерфейса и адреса следующего маршрутизатора, соответствующих этому адресу назначения; обновление заголовка пакета и передача его на блок коммутации. На рис. 3 представлены функции, выполняемые на маршрутизаторе при обработке сетевых пакетов [5]. Эти функции разделяются на три уровня: уровень интерфейсов, уровень сетевого протокола (уровень данных) и уровень протокола маршрутизации (уровень управления).



Рисунок 4 – Структура пакета Ethernet

На первом этапе обработки от сетевого пакета отбрасывается заголовок физического уровня (см. рис. 4) и полученные данные передаются следующему уровню. В современных процессорах эта задача реализована аппаратно. Это ускоряет обработку пакета и освобождает разработчика программного обеспечения от необходимости работать с данными на низком уровне.

На следующем этапе анализируется заголовок IP-пакета (см. рис. 5). Его формат является стандартной структурой [11] и для его разбора необходимы применение битовых операций обработки данных.

0		8		16		31	
Версія	Длина заголовка	Тип сервиса		Общая длина			
Идентификатор пакета				Флаги	Смещение фрагмента		
Время жизни	Протокол		Контрольная сумма заголовка				
Адрес источника							
Адрес назначения							
Параметры и выравнивание							
Данные							

Рисунок 5 – Структура IP-пакета

Длина заголовка занимает четыре бита и содержит количество 32-битных слов в заголовке. Это значение варьируется в диапазоне 5 – 15 слов, что соответствует 20 и 60 байтам.

Поле «Тип сервиса» состоит из восьми бит и определяет правила обслуживания пакета на маршрутизаторе: приоритет пакета, критерий выбора маршрута. Здесь может быть определена передача с минимальной задержкой, максимальной пропускной способностью и наивысшей надежностью маршрута.

Поле общей длины занимает 16 бит и содержит общую длину пакета вместе с заголовком и данными.

Идентификатор пакета (2 байта) используется при сборке фрагментированных пакетов. Для всех пакетов, появившихся в результате фрагментации исходного пакета значение этого поля должно быть одинаковым.

Флаги (3 бита) используются для управления и индикации состояния фрагментации.

Смещение фрагмента (13 бит) используется при сборке фрагментированных пакетов. Оно содержит смещение в байтах текущего фрагмента от начала исходного пакета.

Однобайтовое поле TTL (Time To Live, время жизни) определяет число переходов, которые может осуществить пакет перед тем, как будет уничтожен.

Поле протокола (8 бит) верхнего уровня указывает, данные какого протокола верхнего уровня несет IP-пакет.

Контрольная сумма заголовка занимает два байта и используется для проверки корректности заголовка пакета.

Адреса источника и назначения определяют узел-отправитель пакета и узел, которому он предназначен. Оба поля имеют длину 32 бита, которые однозначно идентифицируют узел во всемирной компьютерной сети.

Поле параметров обычно используется только во время отладки сети. И в большинстве случаев оно остается пустым.

Длина заголовка должна быть кратна 32 битам. Если этого не происходит, то используется поле выравнивания для дополнения заголовка до требуемого размера.

Во время обработки IP-пакета на узле компьютерной сети в первую очередь проверяется

время жизни пакета (поле TTL), используемое для поиска и уничтожения «заблудившихся» пакетов. Это значение задается при создании пакета и означает число маршрутизаторов, через которые может пройти пакет на пути своего следования. При прохождении каждого маршрутизатора TTL уменьшается на единицу и при достижении нуля пакет уничтожается. Пакет уничтожается и в случае, если рассчитанная контрольная сумма заголовка не соответствует содержащийся в пакете.

После первого этапа для пакета ищется путь, по которому он должен быть отправлен далее. Соответствие адресов назначения и путей, по которым пакет должен быть отправлен, хранятся в маршрутной таблице. Структура маршрутной таблицы зависит от используемого алгоритма маршрутизации, но в любом алгоритме каждому отдельному адресу назначения соответствует только одна пара выходной интерфейс – адрес следующего маршрутизатора. Для поиска в маршрутной таблице могут использоваться различные алгоритмы, реализованные аппаратно или программно, многие из них используют бинарные деревья или другие способы оптимизации поиска.

На последнем этапе меняется значение поля TTL, пересчитывается контрольная сумма и полученный заголовок объединяется с телом пакета.

Расчет характеристик сетевого процессора

В предыдущем разделе были описаны три основных этапа обработки пакета на маршрутизаторе. Предложенная модель позволяет гибко распределять приложения между вычислительными ядрами и вычислительные ядра между этапами конвейера. В качестве примера разместим каждый этап обработки на отдельном этапе конвейера.

Допустим, что каждый этап конвейера содержит два вычислительных ядра. В таблице 1 представлены все параметры, необходимые для расчета характеристик сети СМО.

Тогда система уравнений (5) примет вид:

$$\begin{cases} \lambda_0 = \lambda_3 P_{f,3} \\ \lambda_1 = \lambda_0 + \lambda_1 (1 - P_{m,1}) (1 - P_{f,1}) + \lambda_{ram} P_{r,1} \\ \lambda_2 = \lambda_1 P_{f,1} + \lambda_2 (1 - P_{m,2}) (1 - P_{f,2}) + \lambda_{ram} P_{r,2} \\ \lambda_3 = \lambda_2 P_{f,2} + \lambda_3 (1 - P_{m,3}) (1 - P_{f,3}) + \lambda_{O3V} P_{r,3} \\ \lambda_{ram} = \sum_{i=1}^3 \lambda_i P_{m,i} (1 - P_{f,i}) \end{cases} \quad (8)$$

Разделив обе части каждого уравнения из (8) на λ_0 , получим систему для коэффициентов посещений $\alpha_i, i = \overline{1, n}$ и α_{ram} :

$$\begin{cases} 1 = \alpha_3 P_{f,3} \\ \alpha_1 = 1 + \alpha_1(1 - P_{m,1})(1 - P_{f,1}) + \alpha_{ram} P_{r,1} \\ \alpha_2 = \alpha_1 P_{f,1} + \alpha_2(1 - P_{m,2})(1 - P_{f,2}) + \alpha_{ram} P_{r,2} \\ \alpha_3 = \alpha_3 P_{f,2} + \alpha_3(1 - P_{m,3})(1 - P_{f,3}) + \alpha_{ram} P_{r,3} \\ \alpha_{ram} = \sum_{i=1}^3 \alpha_i P_{m,i}(1 - P_{f,i}) \end{cases} \quad (9)$$

Таблиця 1 – Параметри моделюваного мережевого процесора

Параметр	Значення	Описання
μ_1, μ_2, μ_3	от $5 \cdot 10^7$ до $6 \cdot 10^8$	Производительность вычислительных ядер, выраженная в количестве команд, выполняемых в секунду
μ_{OZU}	от $5 \cdot 10^7$ до $1 \cdot 10^9$	Производительность блока ОЗУ, выраженная в количестве обрабатываемых за секунду запросов
P_{f1}	1/201	Для разбора заголовка пакета требуется в среднем 200 команд
P_{f2}	1/501	Для поиска в маршрутной таблице требуется в среднем 500 команд
P_{f3}	1/101	На сборку пакета 100 команд
$P_{ic,1},$ $P_{dc,1}$	0.001 0.01	Вероятности промахов кэша команд и данных для ядра 1, выполняющего разбор заголовка пакета
$P_{ic,2},$ $P_{dc,2}$	0.15 0.25	Вероятности промахов кэша команд и данных для ядра 2, выполняющего поиск в маршрутной таблице
$P_{ic,3},$ $P_{dc,3}$	0.001 0.01	Вероятности промахов кэша команд и данных для ядра 3, выполняющего сборку пакета
$P_{d,1}$	0.5	Вероятность обращения к данным в ОЗУ для ядра 1, выполняющего разбор заголовка пакета
$P_{d,2}$	0,036	Вероятность обращения за данными в ОЗУ для ядра 2, выполняющего поиск в маршрутной таблице
$P_{d,3}$	0.5	Вероятность обращения за данными в ОЗУ для ядра 3, выполняющего сборку пакета

Решив систему (9), получаем значения коэффициентов посещения для всех узлов сети

массового обслуживания, моделирующую маршрутизатор (см. рис. 2):

$$\begin{aligned} \alpha_1 &\approx 205 & \alpha_2 &\approx 205 \\ \alpha_3 &\approx 101 & \alpha_{ram} &\approx 34 \end{aligned} \quad (10)$$

Используя полученные коэффициенты посещения α_i , можно определить характеристики сети СМО.

Величина максимальной пропускной способности маршрутизатора $\lambda_{0,max}$ определяется из условия существования установившегося режима для сети массового обслуживания:

$$\lambda_{0,max} = \min \left(\frac{s_{ram} \mu_{ram}}{\alpha_{ram}}, \min_{i=1,n} \left(\frac{s_i \mu_i}{\alpha_i} \right) \right) \quad (11)$$

На рисунке 6 показана зависимость $\lambda_{0,max}$ от тактовой частоты вычислительных ядер.

В рассматриваемом случае $n = 3$, s_i, μ_i, s_{ram} и μ_{ram} приведены в таблице 2, а α_i и α_{ram} определены из решения системы (9). Из рисунка 6 видно, что максимальная пропускная способность маршрутизатора растет при увеличении тактовой частоты и достигает устойчивого значения при $f_{pr} = 250$ МГц. Следовательно, при высокой производительности вычислительных ядер, общая пропускная способность системы ограничивается производительностью ОЗУ. Поэтому, для увеличения пропускной способности маршрутизатора нужно уменьшить время отклика ОЗУ или реализовать в ОЗУ параллельную обработку нескольких запросов.

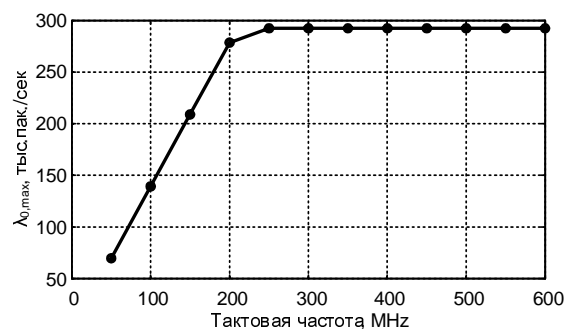


Рисунок 6 – Зависимость максимальной пропускной способности маршрутизатора от тактовой частоты вычислительных ядер, $\mu_{ram} = 10^7$ запросов/сек

На рисунке 7 показано влияние пропускной способности ОЗУ (μ_{ram}) на пропускную способность маршрутизатора при тактовой частоте вычислительных ядер равной 600 МГц. Из графика видно, что производительность СП линейно зависит от пропускной способности ОЗУ, что подтверждает предположение о том, что ОЗУ является узким местом в системе.

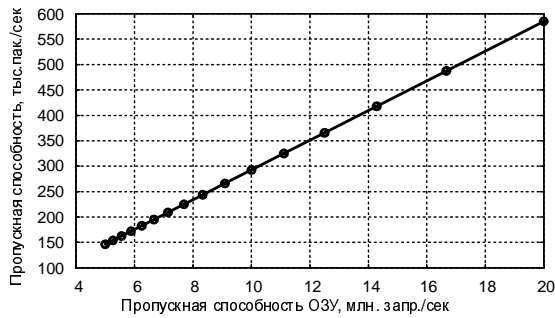


Рисунок 7 – Зависимость максимальной пропускной способности маршрутизатора от пропускной способности ОЗУ (тактовая частота процессора равна 600 MHz)

Загрузка блока определяется как среднее количество занятых приборов:

$$\bar{s} = s - \bar{\rho} \quad (12)$$

где, s – число обслуживающих приборов в блоке, а $\bar{\rho}$ – среднее количество свободных приборов, которое вычисляется как

$$\bar{\rho} = \sum_{i=0}^{s-1} (s-i)P_i \quad (13)$$

где, P_i – вероятность пребывания в системе i заявок.

Рисунок 8 демонстрирует зависимость загрузки блоков (вычислительных ядер и ОЗУ) от интенсивности входного потока при тактовой частоте вычислительных ядер, равной 600 MHz, и ОЗУ, обрабатывающем 10 млн. запросов в секунду.

Из графиков видно, что зависимость носит линейных характер и, при этих характеристиках, ни для одного из блоков она не превышает единицу.

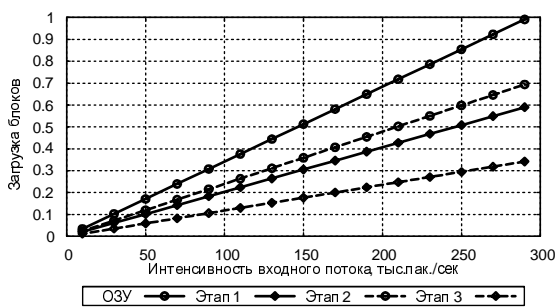


Рисунок 8 – Загрузка блоков маршрутизатора в зависимости от интенсивности входного потока

На рисунке 9 показано влияние тактовой частоты вычислительных ядер на загрузку блоков маршрутизатора. При увеличении тактовой частоты увеличивается и пропускная способность вычислительных ядер, что уменьшает их загрузку.

Влияние пропускной способности ОЗУ на загрузку блоков маршрутизатора показано на рисунке 10. Тактовая частота вычислительных ядер равна 600 MHz.

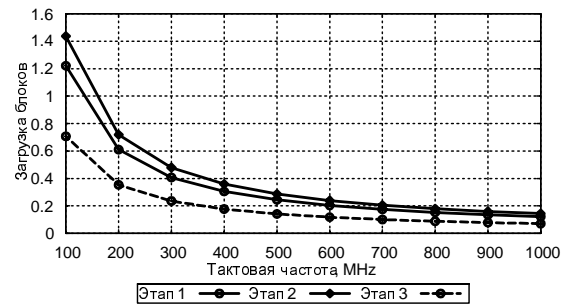


Рисунок 9 – Загрузка блоков маршрутизатора в зависимости от тактовой частоты вычислительных ядер

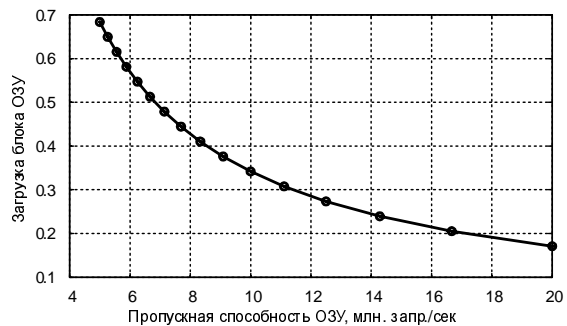


Рисунок 10 – Загрузка блока памяти в зависимости от пропускной способности ОЗУ

Средняя длина очереди определяется по формуле:

$$\bar{v} = \frac{(\lambda/\mu)^{s+1}}{(s-1)!(s-(\lambda/\mu))^2} P_0 \quad (14)$$

где, P_0 – вероятность отсутствия требований в системе.

Зависимость средних длин очередей для блоков сетевого процессора показана на рис. 11 и 12. Из рисунков видно, что при тактовой частоте 600 MHz и пропускной способности ОЗУ, равной 10 млн. запросов/сек., очереди у этапов конвейера очень малы и только для блока ОЗУ очередь достигает больших размеров и уже при интенсивности входного потока равной 270 тыс. пак./сек превышает количество вычислительных ядер в системе.

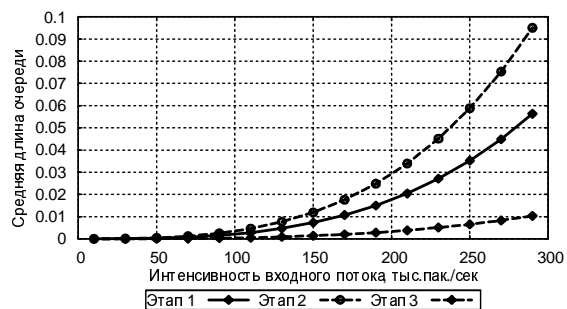


Рисунок 11 – Зависимость средних длин очередей этапов конвейера от интенсивности входного потока

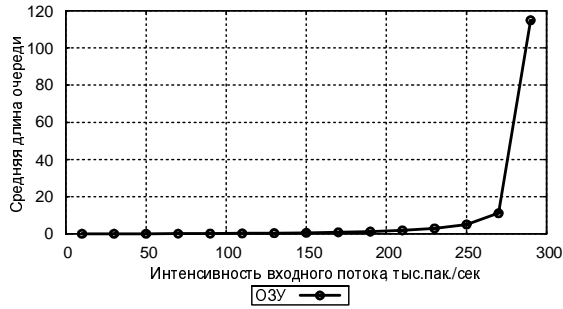


Рисунок 12 – Зависимость средних длин очереди к блоку ОЗУ от интенсивности входного потока

Зависимость средних длин очередей от производительности блоков СП показана на рисунках 13 (для вычислительных ядер) и 14 (для блока ОЗУ). На рисунке 13 показаны характеристики вычислительных ядер при μ_{ram} равному 10 млн. запросов в секунду, а на рисунке 14 – при тактовой частоте 600 МГц. Из рисунков видно, что при увеличении пропускной способности длина очередей быстро падает и стремится к нулю.



Рисунок 13 – Зависимость средних длин очередей от тактовой частоты вычислительных ядер

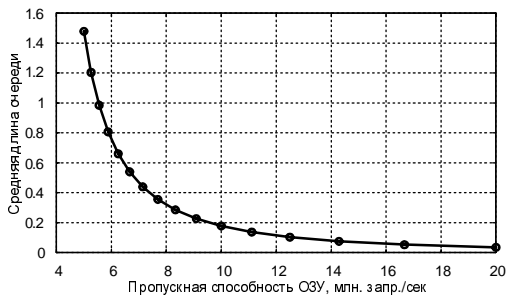


Рисунок 14 – Зависимость средней длины очереди блока ОЗУ от его пропускной способности

Среднее время пребывания заявки в узле вычисляется по формуле:

$$\bar{u} = \frac{\bar{n}}{\lambda} \quad (15)$$

где \bar{n} – среднее число заявок в узле, которое зависит от средней длины очереди, интенсивности входного потока и интенсивности обслуживания:

$$\bar{n} = \bar{v} + \frac{\lambda}{\mu} \quad (16)$$

На рис. 15 и 16 показана зависимость

среднего времени пребывания заявки в узле сети СМО от интенсивности входного потока. Графики строились для тактовой частоты вычислительных ядер, равной 600 МГц и пропускной способности блока ОЗУ — 10 млн. запросов в секунду.

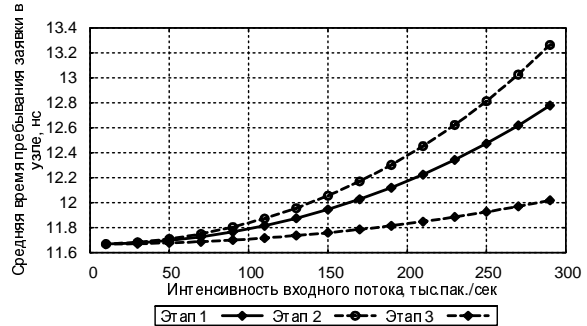


Рисунок 15 – Зависимость среднего времени пребывания пакета на этапах конвейера от интенсивности входного потока

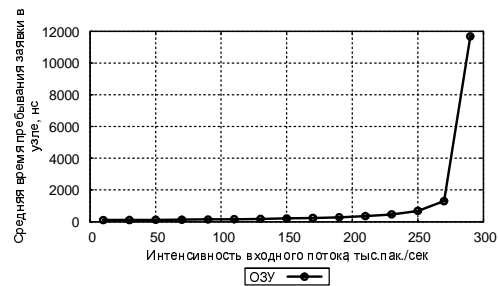
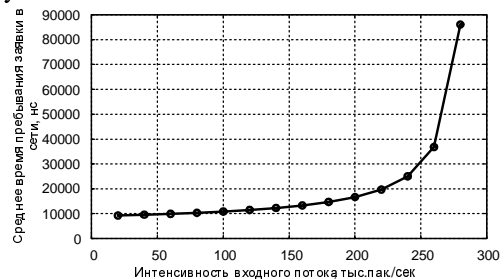


Рисунок 16 – Зависимость среднего времени пребывания пакета в блоке ОЗУ от интенсивности входного потока

Общее время обработки пакета на маршрутизаторе (u_t), зависит от времени обработки в каждом узле (u_i) и коэффициентов посещения узлов (α_i):

$$\bar{u}_t = \alpha_{ram} \bar{u}_{ram} + \sum_{i=1}^n \alpha_i \bar{u}_i \quad (17)$$

На рисунках 17 – 19 представлены графики зависимости общего времени обработки пакета на маршрутизаторе от интенсивности входного потока, тактовой частоты вычислительных ядер и пропускной способности блока



ОЗУ.

Рисунок 17 – Зависимость времени обработки пакета на маршрутизаторе от интенсивности входного потока

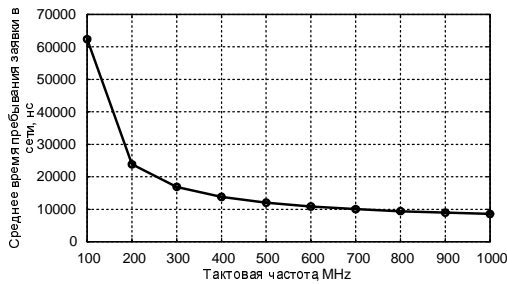


Рисунок 18 – Зависимость времени обработки пакета на маршрутизаторе от тактовой частоты вычислительных ядер

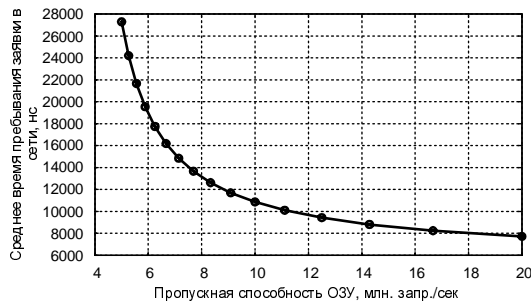


Рисунок 19 – Зависимость времени обработки пакета на маршрутизаторе от пропускной способности блока ОЗУ

При увеличении интенсивности входного потока время пребывания заявки в системе

увеличивается экспоненциально. Зависимость времени пребывания заявки в системе от тактовой частоты вычислительных ядер и производительности блока ОЗУ аналогична зависимости длин очередей от тех же параметров. Это объясняется тем, что большую часть времени в системе пакеты проводят в очередях и, в ситуациях, когда очереди не образуются, пакеты обрабатываются значительно быстрее, чем в ситуациях, когда длины очередей достигают больших значений.

Выводы

Предложенная модель сетевого процессора предоставляет широкие возможности для исследования его функционирования. Она учитывает особенности многоядерной структуры современных сетевых процессоров, позволяет гибко варьировать технические характеристики блоков СП и приложений сетевой обработки данных.

Проведенные эксперименты показали, что оперативная память является узким местом в маршрутизаторе. Поэтому исследование влияния различных способов организации ОЗУ на производительность маршрутизатора является перспективным направлением дальнейших работ.

Литература

1. Chris Rosewarne. Network Processors: Evaluating Architectures for Leading Edge Applications. White Paper. Calyptech. 19th March 2004.
2. N. Shah, "Understanding network processors," Tech. Rep. Version 1.0, EECS, University of California, Berkeley, September 2001.
3. Ладиженський Ю.В. Грищенко В.И. Моделирование сетевых процессоров пакетной обработки данных. Матеріали міжнародної науково-практичної конференції „Інтернет – Освіта – Наука - 2006”, м. Вінниця, 10 – 14 жовтня 2006 р., т. 2, стр. 417-422.
4. Samarjit Chakraborty, Simon Kunzli, Lothar Thiele et. al. Performance evaluation of network processor architectures: combining simulation with analytical estimation. Computer Networks: The International Journal of Computer and Telecommunications Networking. Vol. 41, Iss. 5 (Apr. 2003). 2003. pp. 641 – 665.
5. M. Gries, C. Kulkarni, C. Sauer, K. Keutzer. Comparing Analytical Modeling with Simulation for Network Processors: A Case Study. Design Automation and Test in Europe (DATE), Munich, Germany, March 2003.
6. Ning Weng and Tilman Wolf, "Pipelining vs. multiprocessors – choosing the right network processor system topology," in Proc. of Advanced Networking and Communications Hardware Workshop (ANCHOR 2004) in conjunction with The 31st Annual International Symposium on Computer Architecture (ISCA 2004), Munich, Germany, June 2004.
7. Tilman Wolf, Mark A. Franklin, "Performance Models for Network Processor Design," IEEE Transactions on Parallel and Distributed Systems, vol. 17, no. 6, pp. 548-561, Jun., 2006.
8. M. Ahmadi, S. Wong, A Performance Model for Network Processor Architectures in Packet Processing Systems, Proceedings of the 19th International Conference on Parallel and Distributed Computing and Systems (PDCS 2007), pp. 176-181, Cambridge, Massachusetts, USA, November 2007
9. F. Baker et al. RFC1812 - Requirements for IP Version 4 Routers. Cisco Systems. June 1995.
10. В. Г. Олифер, Н. А. Олифер. Компьютерные сети. Принципы, технологии, протоколы. 3-е изд.-2006, СПб, Изд. дом "Питер", 958 стр.
11. RFC791 - Internet Protocol. DARPA internet program: protocol specification. September 1981.

Поступила в редакцию 30.03.2010