УДК 621.391

Интеллектуальное ядро для сетей связи пятого поколения и Тактильного интернета на базе программно-конфигурируемых сетей

А. А. Атея, аспирант кафедры сетей связи и передачи данных СПбГУТ; eng.abdelhamied@hotmail.com
А.С. Мутханна, доцент кафедры сетей связи и передачи данных СПбГУТ, к.т.н.; ammarexpress@gmail.com
А.Е. Кучерявый, профессор, заведующий кафедрой сетей связи и передачи данных СПбГУТ, д.т.н.; akouch@mail.ru

Аннотация.

Программно-конфигурируемые сети SDN (Software Defined Network) – достаточно новая технология телекоммуникаций, которая с появлением концепции сетей связи пятого поколения приобрела для них исключительно важное значение для решения проблем управления трафиком и уменьшения задержки. В статье предлагается и исследуется новая структура ядра сети на основе технологии SDN для сетей 5G и Тактильного Интернета. Для оптимизации построения такой сети предложен алгоритм, основанный на рое сальп.

Ключевые слова:

Программно-конфигурируемые сети, роевые алгоритмы, контроллер, SDN, 5G, интеллектуальное ядро сети.

Введение. Уже прошло три года с тех пор, как в журнале “Электросвязь” была напечатана первая в стране научная статья по Тактильному Интернету [0]. С тех пор появилось понимание того, что именно эта технология лежит в основе преобразования сетей связи и реализации задач цифровой экономики [14]. Были разработаны предложения по построению сети для Тактильного Интернета на основе использования технологии мобильных граничных вычислений MEC (Mobile Edge Computing) [3]. В данной статье эта методология развита и на построение ядра сети.

Основной проблемой является обеспечение задержки из конца в конец величиной в 1мс. С целью уменьшения круговой задержки и выполнения требования по величине задержки в 1мс для Тактильного Интернета, число сетевых узлов, включенных в процесс обеспечения связи, должно быть как можно меньше и располагаться эти узлы должны как можно ближе к пользовательскому оборудованию. Такие характеристики могут быть достигнуты совместным использованием новых технологий программно-конфигурируемых сетей SDN, виртуализации сетевых функций NFV и мобильных граничных вычислений MEC в сетях связи пятого поколения 5G.

При этом, совместное использование SDN и NFV обеспечивает возможность реализации такой важной концепции как сетевой слайсинг. Сетевой слайсинг можно рассматривать как группу сетевых функций, работающих вместе с определенной технологией радиодоступа (RAT) для достижения рационального варианта использования сети [11]. Другими словами, сетевой слайсинг является способом поддержки услуг связи посредством специального соединения. Сетевой слайсинг позволяет оператору сети построить несколько логических сетей (каждая для определенного варианта использования) на одной физической инфраструктуре [11].

Динамические изменения в объеме поступающего трафика, в свою очередь, приводят к изменению числа задействованных на сети контроллеров. Когда трафик на сети возрастает, новые контроллеры должны быть активированы, в то время как при снижении поступающего трафика определенное число контроллеров может быть деактивировано. Частота изменения трафика приводит к соответствующему процессу активации и деактивации контроллеров в мульти контроллерной сети SDN. Спящий режим может быть рассмотрен как альтернативный вариант деактивации контроллера [12]. Решение об использовании того или иного режима для каждого из контроллеров способствует лучшей производительности системы

в целом и может быть оптимизировано с использованием мета эвристических алгоритмов.

Для мульти контроллерной сети SDN задачи определения числа контроллеров и их размещения по сети могут быть рассмотрена как NP – сложная проблема [13].

В статье будет предложена структура сети для 5G и Тактильного Интернета, в которой технология SDN используется в ядре сети и при этом доступной является технология NFV. В основе предложенных решений лежит также технология мобильных граничных вычислений MEC.

Для оценки производительности сети на базе технологии SDN для сетей связи пятого поколения 5G и Тактильного Интернета в статье оценивается производительность сети. Для начала смоделировано интеллектуальное ядро сети с централизованным контроллером, а затем и мульти контроллерная SDN сеть.

Интеллектуальное ядро сети для 5G и Тактильного Интернета на базе централизованного контроллера. Предоставление услуг Тактильного Интернета требует от сети очень высокой надежности и доступности, а также ультра малой задержки. Это накладывает большие ограничения на компоненты системы с точки зрения контекста, контента и мобильности. При этом Тактильный Интернет требует круговой задержки величиной в 1мс.

На рис.1 показана предлагаемая архитектура сети для выполнения требований Тактильного Интернета по круговой задержке. Сеть предоставления услуг Тактильного Интернета из конца в конец включает в себя пользовательские устройства, сети радиодоступа RAN (базовые станции eNBs), облака, коммутаторы доступа, коммутаторы OpenFlow, промежуточные устройства (Middlebox) и, наконец, SDN контроллер.

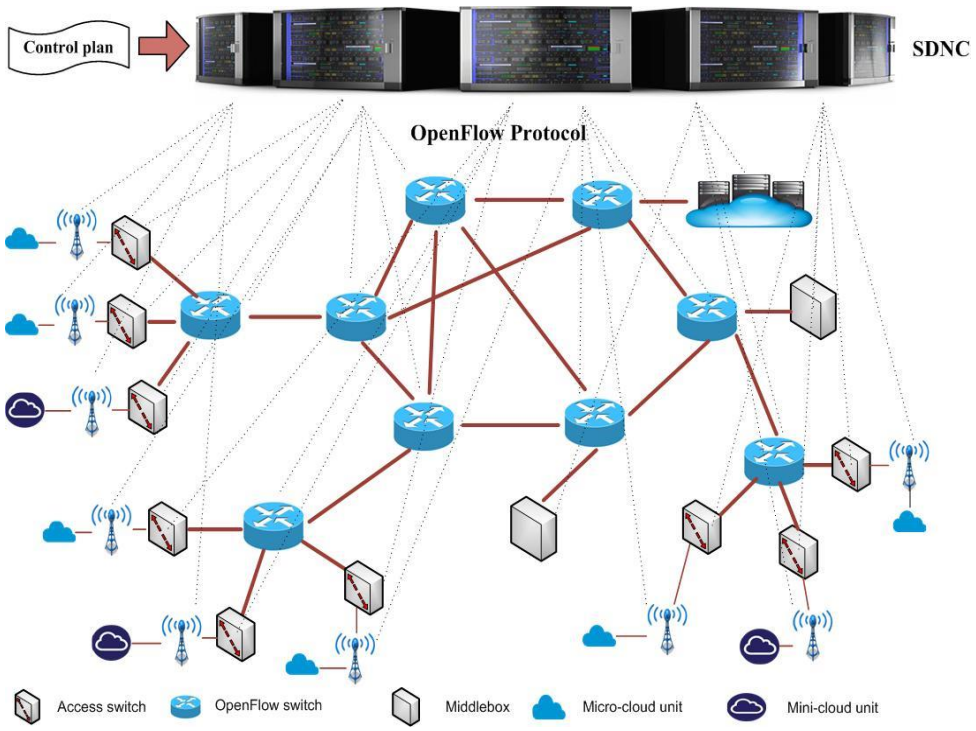

Рис.1. Архитектура ядра сети для выполнения требований Тактильного Интернета.

Каждая базовая станция eNB подсоединена к сети через коммутатор доступа, который обеспечивает классификацию пакетов от пользовательских устройств. Эти коммутаторы реализуются программно и известны как Open vSwitch .

Предлагаемая архитектура сети обладает следующими преимуществами по сравнению с традиционной архитектурой:

1- Сеть становится более гибкой, поскольку для маршрутизации может быть использован централизованный контроллер SDNC.
2- Сеть становится более надежной, используя протоколы открытых систем и другие стандартизованные технологии.
3- Функции виртуализации сети могут быть легко добавлены вследствие использования контроллера SDNC.
4- Требования по достижению определенного уровня качества обслуживания (QoS) могут быть обеспечены при использовании технологии программно-конфигурируемых сетей SDN.

5- Обеспечивается большая масштабируемость сети по сравнению с традиционным пакетным ядром. Действительно, контроллер занимается исключительно процессами управления и не принимает участия в непосредственной передаче данных.

Модифицированное ядро сети на базе технологии SDN с использованием мульти контроллеров. Математическая модель мульти контроллеров SDN.

Предложенная выше архитектура, несмотря на ряд существенных преимуществ по сравнению с существующей, является плохо масштабируемой. С целью устранения этого недостатка далее предложим мульти контроллерную сеть SDN для реализации ядра сети связи пятого поколения и Тактильного Интернета.

Для того, чтобы сформулировать задачу оптимизации, прежде всего, рассмотрим модель такой сети. Мульти контроллер SDN управляет M коммутаторами, которые используют соответствующий интерфейс SDN, например, OpenFlow. Коммутаторы OpenFlow соединены с N контроллерами, распределенными по сети. Обозначим набор контроллеров как C_i (i = 1,2,…,N), а набор коммутаторов как S_j (j = 1,2,…,M).

Для выполнения любой сетевой операции каждый коммутатор OpenFlow запрашивает обслуживание у соответствующего контроллера. Контроллер передает на коммутатор требования по коммутации и таблицу переадресации. Поскольку ресурсы контроллера, включая память, пропускную способность и обработку информации, естественно, ограничены, контроллер Ci может управлять только ограниченным числом коммутатров OpenFlow Ki. Ki - это число коммутаторов, связанных с определенным контроллером Ci в конкретный момент времени, которое может определено из матрицы коммутаторов мульти контроллера путем суммирования числа коммутаторов в ряду контроллера. Пример матрицы коммутаторов мульти контроллерной сети с N = 5 и M = 7 представлен формулой (1), где матрица K включает в себя

число подключенных коммутаторов к каждому из контроллеров. Строки матрицы представляют разные контроллеры, а столбцы - карты коммутаторов. Контроллеры не должны быть перегружены. Это связано с тем, что вероятность сбоя значительно увеличивается с увеличением нагрузки на контроллер, особенно в условиях перегрузки.

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \square\square \; [K] = \begin{bmatrix} 2 \\ 1 \\ 2 \\ 1 \\ 1 \end{bmatrix} \qquad (1)$$

Одним из способов оценки производительности контроллера может служить длительность времени ответа контроллера на запрос коммутатора, что оказывает решающее влияние на задержку [15]. Мульти контроллер может быть представлен моделью с очередями вида M/M/s.

Среднее время ответа T_i контроллера C_i определяется как сумма длительности ожидания в очереди и времени обработки, и может быть вычислено с использованием формулы С Эрланга как функция интенсивности поступления λ_i и интенсивности обслуживания μ.

$$T_i(\lambda) = \frac{C\left(s, \frac{\lambda_i}{\mu}\right)}{s\mu_i - \lambda_i} + \frac{1}{\mu} \qquad (2)$$

В (2) C (s, λ/μ) есть вероятность того, что все серверы в системе находятся в состоянии обслуживания и любой поступающий пакет будет поставлен в очередь. Эту вероятность можно вычислить в соответствии с формулой (3).

$$C\left(s, \frac{\lambda}{\mu}\right) = \frac{\left(\frac{(s\rho)^c}{s!}\right)\left(\frac{1}{1-\rho}\right)}{\sum_{k=0}^{s-1} \frac{(s\rho)^k}{k!} + \left(\frac{(s\rho)^c}{s!}\right)\left(\frac{1}{1-\rho}\right)} = \frac{1}{1 + \left(\frac{1}{1-\rho}\right)\left(\frac{s!}{(s\rho)^c}\right)\sum_{k=0}^{s-1} \frac{(s\rho)^k}{k!}} \qquad (3)$$

$$\rho = \frac{\lambda_i}{s.\mu} \qquad (4)$$

Здесь ρ представляет собой загрузку сервера, что указывает на устойчивость системы. Система имеет устойчивое распределение только тогда, когда ρ меньше 1.

Интенсивность поступления заявок λ_i на контроллер C_i может быть вычислена как сумма средних интенсивностей поступающих заявок на коммутатор, соединенный с контроллером.

$$\lambda_i = \sum_{k_i} \lambda_s \qquad (5)$$

Соответственно, средняя нагрузка на контроллер C_i определяется как как среднее число заявок, находящихся в очереди и в процессе обработки. Используя формулу С Эрланга, несложно получить формулу для определения нагрузки на контроллер L_i (6).

$$L_i(\lambda) = s\rho + \frac{\rho}{1-\rho} C\left(s, \frac{\lambda_i}{\mu}\right) \qquad (6)$$

Задача размещения контроллеров на сети с учетом задержки и требуемых затрат.

Задача оптимизации может быть сформулирована следующим образом:

$$Min\, f(N, C, D) \qquad (8)$$

при ограничениях

$$T_i \le \tau\,, \qquad \forall imA \qquad (9)$$

$$U_{lb} \le U_i \le U_{ub}, \forall imA \qquad (10)$$

Здесь f – нелинейная функция числа размещенных контроллеров N, D – средняя задержка между контроллером и присоединенными коммутаторами (вследствие распространения, очередей и обработки) и C – затраты на размещение контроллеров, включая капитальные CAPEX и операционные OPEX. Эта задача относится к классу многоцелевых задач оптимизации с несколькими ограничениями.

Первое ограничение требует, чтобы среднее время ответа T_i контроллера C_i было меньше некоторой пороговой величины τ, которая является заданной.

Это относится ко всем контроллерам из доступного набора контроллеров [A]. ВеличинаC задается в соответствии с существующими требованиями по качеству обслуживания (QoS). Второе ограничение связано с индексом использования каждого из контроллеров, который может принимать значение между нижней границей индекса использования контроллера U_{lb} и верней границей этого индекса U_{ub}. Обе величины U_{ub} и U_{lb} также задаются исходя из требований по обеспечению уровня качества обслуживания QoS системы в целом. Индекс использования контроллера применяется для отражения характеристик использования мощности, памяти и ресурсов обработки информации.

Определим функцию использования системы, которая представляет собой функцию полезности использования различных решений и обеспечивает выбор наилучшего решения. Функция использования системы обычно применяется для представления значений переменных или событий в реальных числах. Она может быть определена следующим образом:

$$u : X \to \mathrm{R} \tag{11}$$

Первой функцией полезности, которая должна быть рассмотрена, является функция использования времени, которая отражает длительность ответа для каждого контроллера. При этом, в принципе, можно использовать любую функцию общих потерь. Будем далее использовать квадратичную функцию потерь, поскольку она с математической точки зрения хороша вследствие симметрии [16]. Функция полезности времени контроллера C_i есть U_T^{Ci} и определяется в соответствии со следующей формулой:

$$U_T^{C_i} = \begin{cases} \alpha + \beta\left(\tau - T_i(\lambda)\right)^2 & , T_i(\lambda) \le \tau \\ 0 & , T_i(\lambda) > \tau \end{cases} \forall i \in A \tag{12}$$

Здесь α и β являются постоянными, при этом их значения не влияют на решение. Первой константе α может быть назначено определенное значение, которое представляет минимальную ненулевую величину времени

использования $U_{T\text{-}thr}$, когда время ответа равно пороговому значению. Уравнения (13) и (14) определяют обе константы:

$$\alpha = U_{T-thr} \qquad \forall U_T \in [0,1] \tag{13}$$

$$\beta = \frac{(1-U_{T-thr})}{\tau^2} \qquad \forall U_T \in [0,1] \tag{14}$$

Для порогового значения времени использования для каждого контроллера в 70%, tвремя использования может быть переопределено как:

$$U_T^{C_i} = \begin{cases} 0.7 + \frac{0.3}{\tau^2}\left(\tau - T_i(\lambda)\right)^2, T_i(\lambda) \le \tau \\ 0 \qquad\qquad\qquad\qquad , T_i(\lambda) > \tau \end{cases} \forall i \in A \tag{15}$$

Другой важной функцией полезности, которая должна быть рассмотрена, является функция полезности затрат, которая отражает затраты на используемые контроллеры. Затраты, в основном, относятся и к CAPEX, и к OPEX, которые требуются для создания сети контроллеров. Квадратичная функция потерь также представляет собой подходящую функцию для отображения использования затрат. Стоимость использования каждого контроллера из набора доступных контроллеров можно определить следующим образом:

$$U_C^{Ci} = \begin{cases} \varphi\left(U_{ub} - U_{ci}\right)^2 \forall\ U_{ci} \in [U_{lb}, U_{ub}] \\ 0 \qquad\qquad \forall\ U_{ci} \notin [U_{lb}, U_{ub}] \end{cases} \forall i \in A \tag{16}$$

Здесь Ф – постоянная, которая не влияет на решение; независимо от того, какое значение ей присвоено. Подходящее значение для Ф может быть определено в соответствии с (3.17) при значениях U_C между 0 и 1.

$$\varnothing = \frac{1}{\left(U_{ub} - U_{lb}\right)^2} \qquad \forall U_C \in [0,1] \tag{17}$$

Обе функции полезности представляют общую полезность каждого контроллера в сети. Естественно, при сложении обеих функций полезности могут использоваться соответствующие веса. Уравнение (18) показывает общую полезность для каждого из контроллеров U_{Ci}.

$$U_{ci} = \delta_C U_C^{Ci} + \delta_T U_T^{Ci} \quad \forall i \in A \tag{18}$$

Здесь δ_C – весовой коэффициент для затрат и δ_T – весовой коэффициент для времени. Общая функция использования для системы представляет собой среднюю величину использования для каждого из контроллеров и может быть вычислена как:

$$U = \sum_A U_{ci} \Big/ |A| \tag{19}$$

Исходя из сказанного, общие функции полезности для затрат U_{CT} и для времени U_{TT} для всех доступных контроллеров могут быть вычислены следующим образом:

$$U_{CT} = \sum_A U_C^{Ci} \Big/ |A| \tag{20}$$

$$U_{TT} = \sum_A U_T^{Ci} \Big/ |A| \tag{21}$$

Алгоритм CSSA для оптимального числа контроллеров и их размещения на сети. В алгоритме SSA определение местоположения корма, которое является лучшей позицией для сальпы, является решением оптимизационной задачи. В модели, используемой в работе, необходимо найти оптимальное число контроллеров и оптимальные взаимосвязи контроллера с каждым из коммутаторов из набора коммутаторов S. Эта задача рассматривается как NP-сложная. Для решения такой задачи могут быть использованы мета эвристические алгоритмы, поскольку детерминированные использовать эффективно не удается. Применим алгоритм SSA, как один из роевых алгоритмов, для решения задачи размещения контроллеров и нахождения их оптимальных взаимосвязей. Алгоритм SSA демонстрирует высшую производительность по сравнению с иными алгоритмами роев частиц PSO.

Основная идея алгоритма SSA заключается в итеративном параллельном поиске нескольких сальп для получения оптимального решения. В требуемой в работе задаче оптимальным решением является оптимальное количество контроллеров и оптимальное распределение контроллеров по коммутаторам. Следовательно, два алгоритма во вложенном цикле должны быть

использованы для получения наилучших решений. Будем использовать два алгоритма, основанных на хаотическом алгоритме CSSA. Алгоритм 1 использует псевдокод для CSSA, введенный для определенной проблемы, где каждый шаг представляет собой число контроллеров в сети. На выходе этого алгоритма получаем оптимальное число контроллеров. Алгоритм 2 использует псевдокод для CSSA, который применяется для поиска оптимальных соединений для всех коммутаторов на основе оптимального числа контроллеров, определенного в соответствии с алгоритмом 1. Каждый шаг в алгоритме 2 анализирует все доступные соединения для всех коммутаторов с их выделенными контроллерами. Это представляется M-мерным вектором, где каждое измерение соответствует конкретному коммутатору. На выходе второго алгоритма получаем оптимальное распределение коммутаторов по контроллерам.

Система начинает работу с набором исходных параметров для алгоритма CSSA, включая нижнюю границу, верхнюю границу и максимальное число итераций, после чего случайным образом инициализируются n сальп, причем каждая сальпа представляет доступное число контроллеров в сети. Полезность каждой сальпы вычисляется в соответствие с (19) и сальпа с наибольшей полезностью рассматривается как текущее наилучшее решение. Местоположение сальпы с наибольшей полезностью представляется как местоположение корма. Параметры алгоритма SSA перезагружаются в соответствии с местоположением сальп. Процесс перезагрузки местоположения сальп и оценки полезности кажой сальпы повторяется до тех пор, пока оптимальное решение не будет найдено или до максимального числа итераций.

Результаты моделирования для сети с централизованным контроллером. Проанализируем результаты моделирования предложенной структуры ядра сети для реализации услуг Тактильного Интернета в сетях связи пятого поколения 5G.

Существует целый ряд эффективных систем моделирования и оценки производительности и параметров сетей, построенных на основе SDN и MEC. Одной из наиболее эффективных, надежных и масштабируемых систем моделирования является система CloudSim и ее расширения. CloudSim представляет собой систему моделирования на основе Java, которая позволяет моделировать и оценивать облачные структуры [17]. CloudSimSDN - это основанная на Java система моделирования. Эта структура разработана, в основном, для целей моделирования совместного использования технологий SDN и облачных структур.

В CloudSimSDN централизованный контроллер SDNC является программируемым и позволяет тестировать политики управления виртуальными машинами и алгоритмы функционирования планировщика нагрузки [18]. Кроме того, этот централизованный контроллер отвечает и за динамическое управление ресурсами в центрах обработки данных. На основании [19] можно показать, что симулятор CloudSimSDN предоставляет множество функций и возможностей с лучшей или, по крайней мере, такой же производительностью, что и среда моделирования Mininet. Кроме того, использование CloudSimSDN дает большие возможности для модификаций и расширений.

Далее будем анализировать результаты моделирования для двух случаев. В первом случае имеем два микро облака и две базовые станции eNBs, причем обе эти базовые станции соединены с одним и тем же мини облаком. Во втором случае микро облака распределены по разным сотам, и базовые стации соединены с различными мини облаками. Круговая задержка рассматривается как характеристика производительности в анализируемых сетях. Процесс моделирования многократно повторяется с различной полосой пропускания каждый раз, но объем данных остается неизменным в каждом случае. Все основные параметры моделирования приведены в таблице 1.

Таблица 1. Параметры моделирования для интеллектуального ядра сети с централизованным контроллером SDN.

Параметр моделирования	Значение
OpenFlow Switch processing delay	5мкс
SDN Controller processing delay	0.5мкс
Arrival rate of the Micro-cloud unit λi	15
The communication latency inside the cellular cell	100мкс
Bandwidth	Переменная
Память RAM микро облака Память RAM мини облака	1024Mb,1 Gb 2048Mb,5 Gb

На рис.2. и рис.3 показаны результаты для первого и второго случаев моделирования. Как видим, круговая задержка уменьшается с увеличением пропускной способности системы. Для второго случая круговая задержка больше, чем в первом случае. Это связано с тем, что контроллер базовой сети не участвует в процессе соединения, поскольку мини облако управляет этим процессом самостоятельно.

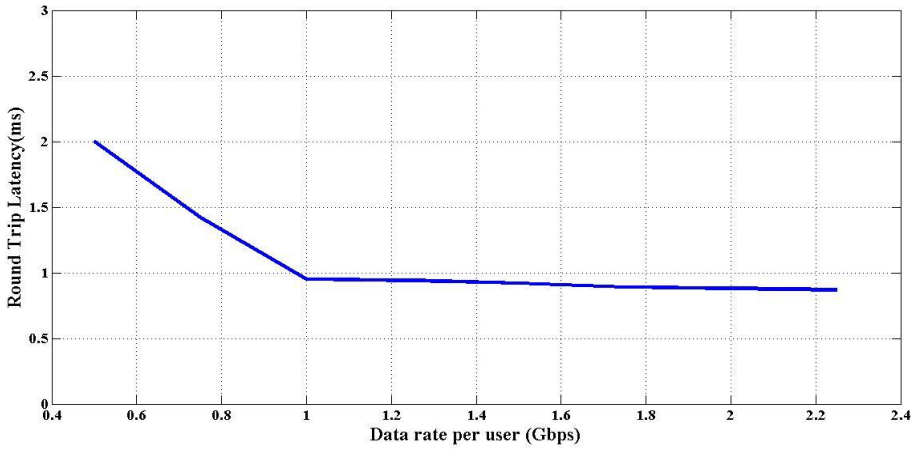

Рис.2. Результаты моделирования для первого случая.

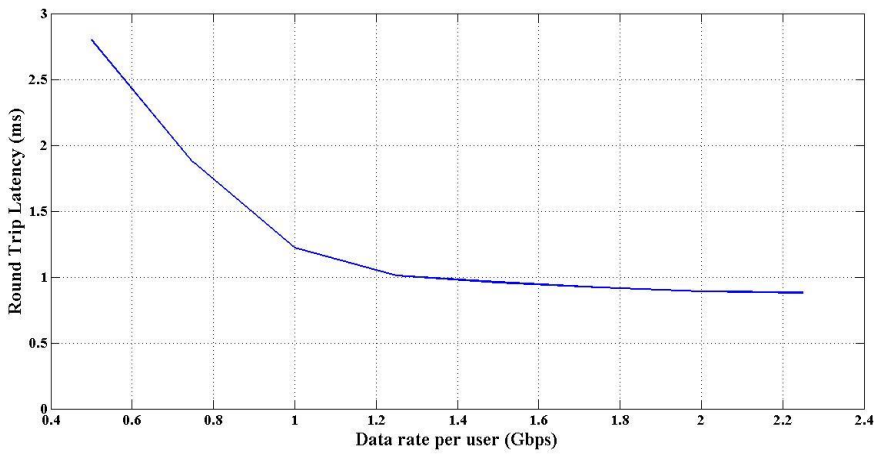

Рис.3. Результаты моделирования для второго случая.

Основываясь на [20], [4], можно считать, что пользовательская скорость для сетей связи пятого поколения 5G будет находиться в пределах от 1Гбит/с до 10Гбит/с. Для наименьшей ожидаемой скорости передачи в битах (1Гбит/с) предложенная в работе архитектура сети для реализации услуг Тактильного Интернета обеспечивает задержку величиной 0,95мс для первого случая моделирования и 1,22мс для второго случая. Кроме того, поскольку скорость передачи данных будет и выше 1Гбит/с, круговая задержка будет меньше полученных значений, что позволяет предложенной в статье архитектуре сети справиться с проблемой обеспечения круговой задержки величиной в 1мс.

Далее, основываясь на версии 13 3GPP, можно видеть, что система LTE достигает двусторонней задержки 16мс для скорости передачи 100Мбит/с и 8мс для скорости 1Гбит/с для версии 15 [21]. Таким образом, предложенное в статье решение система может повысить эффективность сети в плане обеспечения задержки и обеспечить реализацию услуг Тактильного Интернета в сетях связи пятого поколения. По сравнению с традиционным решением по построению ядра сети Evolved Packet Core, основанным на версии 15 3GPP, предложенное в работе решение обеспечивает уменьшение круговой задержки на 88% для первого случая и на 84% для второго случая.

Сравнение алгоритмов оптимизации для мульти контроллерной сети. В этом разделе предложенный в работе алгоритм CSSA для размещения контроллеров SDN сравнивается с другими алгоритмами для решения этой же проблемы с точки зрения оценки производительности и иных характеристик. Три основных алгоритма выбраны для сравнения: игровой алгоритм ненулевой суммы, предложенный [22] и два эффективных роевых интеллектуальных алгоритма. Эти два роевых алгоритма – алгоритм роя частиц PSA (Particle Swarm algorithm) и алгоритм стаи серых волков GWA (Grey Wolf algorithm).

Оптимизация роя частиц PSO (Particle Swarm Optimization) является наиболее общеупотребительной роевой техникой, когда моделируется поведение птиц во время перелетов и охоты [23]. Оптимизация серого волка GWO (Gray Wolf Optimization) является основанной на роевой технике имитацией поведения серых волков во время охотыis [24]. При использовании GWA стая серых волков подразделяется на четыре группы: альфа, бета, дельта и омега. Альфа-волк является лидером стаи и представляет собой наилучшее решение. Бета и дельта волки являются лидерами второго и третьего уровня, и это дает, соответственно, второе и третье лучшие решения задачи. Уравнения, используемые для моделирования и представления алгоритмов PSA и GWA. В качестве показателя производительности, используемого для процесса сравнения между предлагаемым алгоритмом и тремя другими алгоритмами, применяется длительность времени выполнения алгоритма. Эта длительность представляет собой задержку, возникающую вследствие выполнения алгоритма для получения оптимального решения, и считается важной метрикой при оценке алгоритмов оптимизации. Кроме того, эта длительность является критически важной задержкой для сетей SDN, поскольку при значительной ее величине функционирующие контроллеры могут войти в состояние перегрузки.

На рис.4 показаны длительности времени выполнения вычислений для каждого алгоритма для различных значений порога задержки τ. Алгоритм CSSA имеет лучшую производительность с точки зрения длительности

времени для всех значений порога задержки τ. Следующим по эффективности является алгоритм серого GWO, а эффективность использования алгоритма PSO занимает промежуточное место между алгоритмами серого волка и ненулевой суммы.. Наихудший алгоритм с точки зрения вычисления времени - это алгоритм игры с ненулевой суммой. Длительность времени выполнения представляет собой очень важный показатель, так как быстрое выполнение необходимых вычислений обеспечивает лучшую производительность сети и поддерживает требуемый уровень качества обслуживания QoS, кроме того, уменьшается вероятность отказа контроллера.

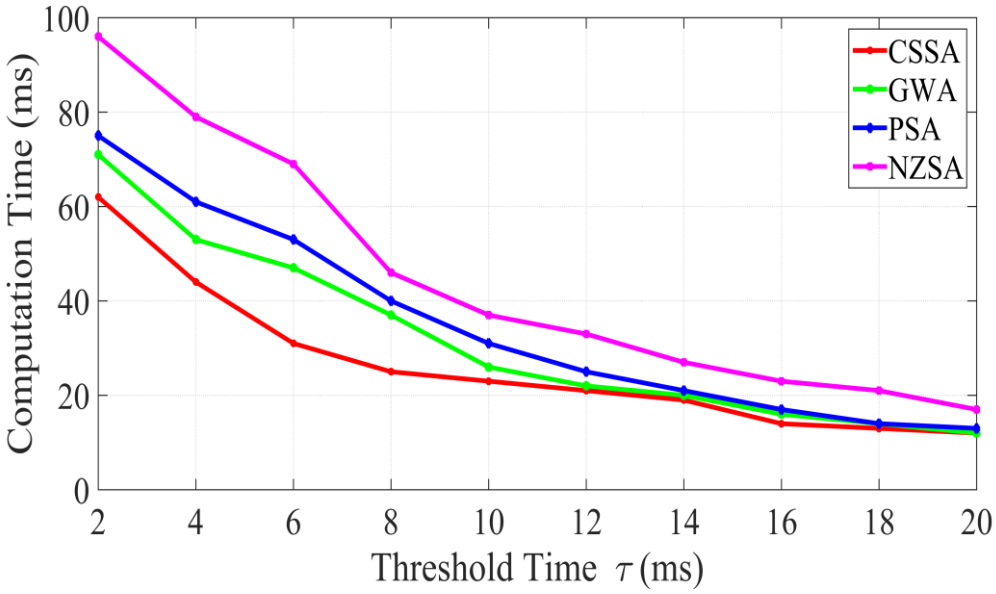

Рис.4. Длительность времени вычисления для кадого алгоритма при различных значения порога задержки.

Кроме того, длительность времени выполнения изменяется с изменением индекса максимальной загрузки контроллера Uub как для предложенного CSSA, так и для трех других анализируемых алгоритмов. На рис.5 показаны результаты моделирования для всех четырех алгоритмов. CSSA достигает наилучшей производительности с точки зрения длительности времени вычислений для всех рассмотренных значений Uub. Важно отметить также и то, что эти три анализируемые алгоритма в некоторых рассмотренных случаях не могут достичь оптимального решения из-за локальных оптимумов, что для алгоритма CSSA обеспечивается с помощью хаотических карт.

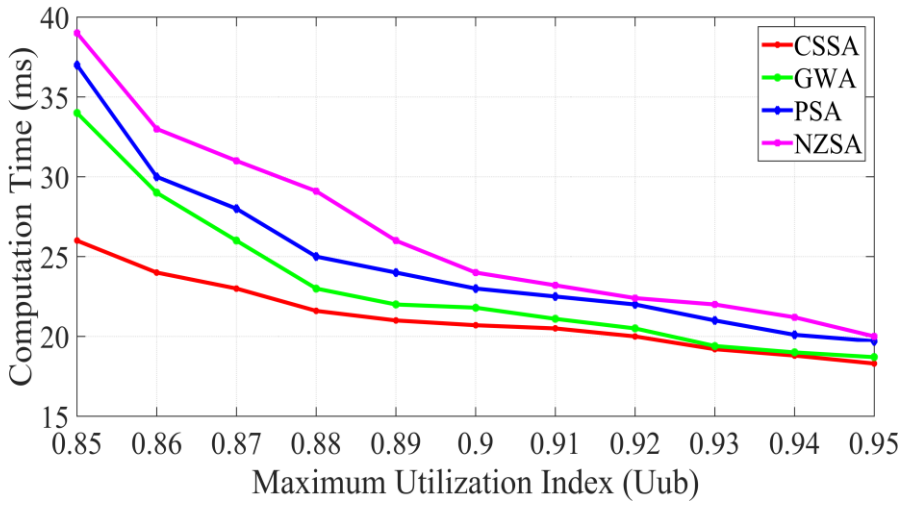

Рис.5. Длительность времени вычисления для каждого алгоритма в зависимости от максимального индекса использования.

Заключение. В статье предложено интеллектуальное ядро сети для реализации сетей связи пятого поколения 5G и приложений Тактильного Интернета на основе использования технологии программно-конфигурируемых сетей SDN. При этом рассмотрены две основные архитектуры сети: с централизованным контроллером и мульти контроллерная сеть. Использование технологии SDN в ядре сети позволяет уменьшить задержку и увеличить производительность сети в целом. Это достигается вследствие эффективного управления со стороны централизованного контроллера SDNC и обеспечения с его помощью гибкой маршрутизации между двумя оконечными точками сети.

Решены задачи по обеспечению задержки совместно с задачами по обеспечению экономически эффективного размещения контроллеров. Сформулирована задача оптимизации и разработан мета эвристический алгоритм для решения задачи оптимизации. В статье проанализирован хаотический алгоритм роя сальп CSSA, который использован для разработки алгоритма оптимизации мульти контроллерной сети SDN для нахождения оптималбного числа контроллеров и оптимального распределения коммутаторов по контроллерам, что позволяет минимизировать задержку и затраты на создание и эксплуатацию сети. Использование хаотических карт

позволяет как увеличить производительность процесса оптимизации, так и предотвращает принятие локальных оптимумов за глобальный.

Список литературы.

1. Атея А.А., Выборнова А.И., Кучерявый А.Е. Многоуровневая облачная архитектура для услуг тактильного интернета, «Электросвязь», №2, 2017.
2. Simsek, M., Aijaz, A., Dohler, M., Sachs, J. and Fettweis, G., 2016. 5G-enabled tactile internet. IEEE Journal on Selected Areas in Communications, 34(3), pp.460-473.
3. Ateya, A.A., Vybornova, A., Kirichek, R. and Koucheryavy, A., 2017, February. Multilevel cloud based Tactile Internet system. In Advanced Communication Technology (ICACT), 2017 19th International Conference on (pp. 105-110). IEEE.
4. Alliance, N.G.M.N., 2017. 5G white paper. Next generation mobile networks, white paper.
5. Гимадинов Р. Ф., Мутханна А. С., Кучерявый А. Е. Кластеризация в мобильных сетях 5G. случай частичной мобильности // Информационные технологии и телекоммуникации. — 2015 . — № 2 (10) . — С. 44-52 . — ISSN: 2307-1303
6. Б.С.Гольдштейн, А.Е.Кучерявый. Сети связи пост-NGN. БХВ, С.Петербург, 2013.
7. Karl, H., Dräxler, S., Peuster, M., Galis, A., Bredel, M., Ramos, A., Martrat, J., Siddiqui, M.S., van Rossem, S., Tavernier, W. and Xilouris, G., 2016. DevOps for network functionvirtualisation: an architectural approach. Transactions on Emerging Telecommunications Technologies, 27(9), pp.1206-1215.
8. A. Ateya, M. Al-bahri, A. Muthanna, and A. Koucheryavy “End-to-end system structure for latency sensitive applications of 5G,” Электросвязь,Vol. 6, pp. 56-61, 2018.
9. Athmiya, N.S., Shobha, K.R. and Sarimela, V., 2016, May. Feasibility study and implementation of openflow based SDN controller for tactical scenario. In

Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE International Conference on (pp. 789-794). IEEE.

10. Shariat, M., Gutierrez-Estevez, D.M., Vijay, A., Safjan, K., Rugeland, P., Silva, I., Lorca, J., Widmer, J., Fresia, M., Li, Y. and Siaud, I., 2016. 5G radio access above 6 GHz. Transactions on Emerging Telecommunications Technologies, 27(9), pp.1160-1167.

11. Кучерявый А. Е., Прокопьев А. В., Кучерявый Е. А. Самоорганизующиеся сети — СПб.: Типография Любавич, 2011. — 312 с.

12. C. Qiu, C. Zhao, F. Xu, and T. Yang, “Sleeping mode of multi-controller in green software-defined networking,” EURASIP Journal on Wireless Communications and Networking, no. 1, p. 282, Dec. 2016.

13. H. Huang, S. Guo, P. Li, B. Ye, and I. Stojmenovic, “Joint optimization of rule placement and traffic engineering for QoS provisioning in software defined network,” IEEE Transactions on Computers, vol. 64, no. 12, pp. 3488–3499, Dec. 2015.

14. Бородин А.С., Кучерявый А.Е. Сети связи пятого поколения как основа цифровой экономики «Электросвязь». 2017. № 5. С. 45-49.

15. E. Sakic, and W. Kellerer, “Response Time and Availability Study of RAFT Consensus in Distributed SDN Control Plane,” IEEE Transactions on Network and Service Management, vol. 15, no. 1, pp. 304–318, Mar.. 2018.

16. P. Franses, R. Legerstee, and R. Paap, “Estimating loss functions of experts,” Applied Economics, vol. 49, no. 4, pp. 386–396, Jan. 2017.

17. Kumar, R. and Sahoo, G., 2014. Cloud computing simulation using CloudSim. arXiv preprint arXiv:1403.3253.

18. CloudSimSDN project https://github.com/jayjmin/cloudsimsdn

19. Son, J., Dastjerdi, A.V., Calheiros, R.N., Ji, X., Yoon, Y. and Buyya, R., 2015, May. Cloudsimsdn: Modeling and simulation of software-defined cloud data centers. In Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on (pp. 475-484). IEEE.

20. Zheng, K., Zhao, L., Mei, J., Dohler, M., Xiang, W. and Peng, Y., 2015. 10 Gb/s hetsnets with millimeter-wave communications: access and networking-challenges and protocols. IEEE Communications Magazine, 53(1), pp.222-231.

21. Lauridsen, M., Gimenez, L.C., Rodriguez, I., Sorensen, T.B. and Mogensen, P., 2017. From LTE to 5G for Connected Mobility. IEEE Communications Magazine, 55(3), pp.156-162.

22. H. K. Rath, V. Revoori, S. M. Nadaf, and A. Simha, "Optimal controller placement in Software Defined Networks (SDN) using a non-zero-sum game," in Proc. IEEE 15th International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM), IEEE, Jun. 2014, pp. 1–6.

23. D. Bratton, and J. Kennedy, "Defining a standard for particle swarm optimization," in Proc. Swarm Intelligence Symposium, SIS 2007, IEEE, 2007, pp. 120–127.

24. H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: a review of recent variants and applications," Neural Computing and Applications, pp. 1–23, 2017.