

УДК 621.391

КОМПЛЕКСНЫЙ МЕТОД ПОВЫШЕНИЯ ДОСТУПНОСТИ СЕРВИСОВ В ПРОГРАММНО- КОНФИГУРИРУЕМЫХ СЕТЯХ, ОСНОВАННЫЙ НА ДИНАМИЧЕСКОЙ РЕПЛИКАЦИИ



[Х.М. Мухи-Алдин](#)

Одесский национальный
политехнический университет

[Е.Б. Ткачова](#), [М.Т. Салах](#)

Харьковский национальный
университет радиоэлектроники

Abstract – The work is devoted to the increasing quality of services in Software-Defined Network by modifying the resource allocation during service provision and services' replication techniques. The well-known replication techniques are analyzed in the work. The result of the analysis shows that existing replication techniques that are used in distributed networks should be revised with the purpose to improve their effectiveness and as a result, increase the availability of services. With the aim to increase the effectiveness of replication techniques in software-defined networking the solution of next tasks are proposed: determining the decision about start replication procedure based on simultaneously increasing of service popularity and degradation of quality of service; determine the required minimum number of replicas and their allocation in the network based on intensive of incoming customers' requests; checking the consistency of new replica of service with almost existing in the network. The complex method of dynamic service replication that is based on this particular solution is proposed. The model of complex services provision is also proposed in the work. The proposed model allows to analytically analyze the value of each monatomic service in complex solution. The case of using the proposed model of the monatomic services that included in complex service can be preplaced by the service with better quality characteristics.

Анотація – У статті запропоновано модель формування та надання комплексного сервісу і метод динамічної реплікації, що дозволяють підвищити доступність сервісів у програмно-конфігурованих мережах. Проведено огляд загальновідомих методів реплікації, які знайшли широке застосування в розподілених мережах, наведено їх основні переваги та недоліки. На підставі результатів проведеного аналізу та з урахуванням особливостей формування сервісів у програмно-конфігурованих мережах запропоновано метод, який включає в себе вирішення наступних завдань: оцінки популярності сервісу та вибору моменту запуску операції реплікації на підставі значення порога реплікації; обчислення оптимальної кількості реплік і їх розміщення на обчислювальних вузлах; перевірки несуперечності нової репліки сервісу із вже існуючими у мережі.

Анотация – В статье предложена модель формирования и предоставления комплексного сервиса и метод динамической репликации, позволяющие повысить доступность сервисов в программно-конфигурируемых сетях. Проведен обзор общеизвестных методов репликации, которые нашли широкое применение в распределенных сетях, приведены их основные преимущества и недостатки. На основании результатов проведенного анализа и с учетом особенностей формирования сервисов предложен метод, который включает в себя решение следующих задач: оценки популярности сервиса и выбора момента запуска операции репликации на основании значения порога репликации; вычисления оптимального количества реплик и их размещения на вычислительных узлах; проверки непротиворечивости новой реплики сервиса уже существующим в сети.

Введение

Рост бизнес-потребностей отдельных пользователей и целых организаций обуславливает популярность и стремительное развитие современных мультисервисных сетей и облачных сетевых решений. Необходимость обеспечения доступности и гарантированного качества предоставляемых сервисов в режиме 24/7 повышает требования к механизмам управления и передачи в таких сетях и обуславливает постоянную потребность в их совершенствовании. В таких условиях провайдеры и крупные

IT-организации для обеспечения требуемого качества обслуживания (Quality of Service, QoS), а также снижения стоимости предоставляемых услуг прибегают к использованию принципов программного конфигурирования при построении мульти-сервисных сетей (так называемой концепции Software-Defined Networking, SDN) [1-3].

Основной идеей SDN является отделение уровня управления сетью (control plane) от уровня передачи данных (forwarding plane) [2, 3]. Вся логика и функции управления переносятся на отдельное устройство – контроллер, обеспечивающее централизованное программно-реализуемое управление на основе оптимальных для сети в целом решений [4, 5]. К основным преимуществам SDN относятся быстрое развертывание, гибкое комбинирование множества сетевых функций на одной серверной платформе, простота и систематизированный характер процедуры администрирования. В соответствии со статистическими данными [6, 7] применение концепции SDN позволяет значительно снизить материальные затраты на такие виды услуг как модернизация сервисов, управление, администрирование и техническая поддержка.

Вместе с тем, в рамках SDN открытым остается вопрос деградации качества сервисов, обусловленной изменением физических характеристик сетевой инфраструктуры. Отказ системы хранения, недостаточность сетевых ресурсов, отказ вычислительных узлов (серверов, физических и виртуальных вычислительных машин) являются основными причинами, вызывающими деградацию QoS в SDN [8]. Как показывает анализ, наиболее эффективными и распространенными решениями быстрого устранения подобных сбоев и восстановления требуемого уровня QoS являются механизмы кластерной [9] и/или глобальной репликации [10], которые заключаются в создании копий запрашиваемого сервиса и их размещении на вычислительных узлах сетевой инфраструктуры. В целом репликация позволяет повысить производительность и надежность сети, а также обеспечить требуемый уровень доступности сервисов при практически любых изменениях в сетевой инфраструктуре [9-11]. При этом управление жизненным циклом, количеством и месторасположением реплик имеет весомое значение. Например, формирование и хранение избыточного количества реплик ведет к повышению накладных расходов, а месторасположение реплики существенно влияет на показатели производительности сети и доступности услуги [11].

I. Обзор известных механизмов репликации, применяемых в распределенных вычислительных сетях

На сегодняшний день множество работ посвящено поиску оптимальных механизмов репликации данных в распределенной сетевой инфраструктуре [9]. При этом могут быть выделены следующие основные стратегии репликации: статическая и динамическая, централизованная и децентрализованная, синхронная и асинхронная репликация, каждая из которых находит свое применение в зависимости от топологии и масштабов сети, а также территориального размещения вычислительных элементов [11-14]. Среди перечисленных методов наибольшую популярность и широкое практическое применение приобрела динамическая асинхронная репликация,

основным преимуществом которой является более низкие объемы передаваемых служебных и пользовательских данных.

Одной из первых работ, посвященных динамической репликации сервисов в распределённых, в частности облачных сетях является работа японских ученых Т. Ikeda, М. Ohara и S. Fukumoto [15]. Особенностью предложенного авторами метода [15] является применение асинхронного режима репликации с возможностью сохранения нескольких экземпляров копий сервиса на вычислительном узле. Операция чтения-записи осуществляется на стороне клиента, при этом вычислительный узел, на котором размещена реплика, не владеет информацией о версии реплики. Выбор определенного экземпляра реплики вычислительным элементом осуществляется на основе ее популярности. Управление и поддержка нескольких экземпляров сервиса в данном случае является ресурсоемкой задачей – количество сервисов, а, следовательно, и количество реплик неуклонно растет, что приводит к иррациональному использованию сетевых ресурсов.

В работе [16] основное внимание уделено вопросам формирования оптимального количества копий и их распределения между вычислительными устройствами. Однако алгоритм репликации, предложенный в этой работе, не позволяет учитывать вероятность и причины отказа в обслуживании. Так, обращения к вычислительному элементу, от которого получен отказ в обслуживании, могут повторяться снова и снова, приводя к потерям пакетов, а, следовательно, деградации качества обслуживания.

В работе [17] предложена методика хранения и формирования реплик, которая позволяет обеспечивать динамическое распределение сетевых ресурсов между несколькими пользователями или приложениями путем анализа популярности и стоимости использования услуг. Распределение сетевых ресурсов между приложениями выполняется динамически и зависит от типа сервиса и требований QoS. Эффективность данного подхода в процессе предоставления высококачественного контента ограничена из-за длительности времени принятия решения о необходимости репликации и низкой точности распределения сетевых ресурсов.

В работе [18] рассмотрены механизмы асинхронной репликации, а также предложена комплексная методика, которая позволяет уменьшить объемы передаваемых данных и нивелировать противоречия между репликами. Однако возникновение большого количества запросов, обращенных к одному и тому же вычислительному устройству одновременно, может привести к его перегрузке и, как следствие, отказу в обслуживании. Кроме этого оригинал услуги или ее основная копия всегда должны быть доступны, что накладывает ряд ограничений на использование предложенного метода.

Two-level Distributed Hash Table метод предложен в [19]. В предлагаемом подходе определяется необходимость формирования реплики на основе информации о ее популярности. Решение о размещении реплики принимается на основе двух критериев: основной локации запросов пользователей на предоставление сервиса и физических характеристик вычислительных узлов.

Приведенные выше подходы нашли широкое применение в процессе управления распределенной сетевой инфраструктурой. Однако их непосредственное применение в SDN затруднительно в силу ряда причин, связанных с высокими затратами сетевых ресурсов в процессе распределения и хранения реплик [15], отсутствием методов переадресации запросов на вычислительный узел, содержащий реплику [17], отсутствием учета производительности вычислительных элементов, на которых размещена реплика, [16] и механизмов балансировки сетевой нагрузки в процессе распределения и использования реплик услуги [18, 19].

Таким образом, в условиях применения репликации сервисов как метода повышения их доступности и обеспечения требуемого качества обслуживания в SDN известные решения должны быть пересмотрены в сторону повышения их эффективности, что обуславливает актуальность задачи управления количеством и расположением реплик, в частности подзадач определения момента принятия решения о формировании реплик сервиса, определения необходимого минимального количества реплик и выбора их месторасположения, а также установления момента принятия решения о непротиворечивости новой реплики сервиса в контексте их применения в SDN.

II. Модель процесса формирования комплексного сервиса в программно-конфигурируемых сетях

Распределенные вычислительные сети, построенные в соответствии с концепцией SDN, зачастую имеют сложную структуру, фрагменты которой географически удалены друг от друга. В соответствии с концепцией SDN, формирование информации о текущей топологии сети на уровне управления осуществляется на основе данных, полученных от множества активных сетевых элементов (OpenFlow коммутаторов) посредством протокола OpenFlow [3, 5, 19]. При этом контроллер обладает не полной картой сетевой инфраструктуры, а множеством логических сетевых объединений - виртуальных локальных сетей (Virtual Local Area Network, VLAN) (рис.1).

SDN-контроллер является ключевым звеном системы управления и занимает центральное место в сетевой архитектуре. При этом, эффективность управления сервисами в SDN всецело зависит от функциональных особенностей контроллера и принципов его взаимодействия с рядом других вспомогательных модулей управления, таких как планировщик задач, балансировщик нагрузки, модуль оркестрации сервисов, брокер репликации и прочих [3].

Процесс предоставления сервисов представляет собой следующую последовательность действий: потребитель сервиса запрашивает определенный набор услуг, требования потребителя посредством OpenFlow коммутатора передаются на контроллер. Контроллер, на основании информации о текущем количестве услуг и их местоположении, формирует ответ, в котором указывает данные для доступа к требуемому сервису.

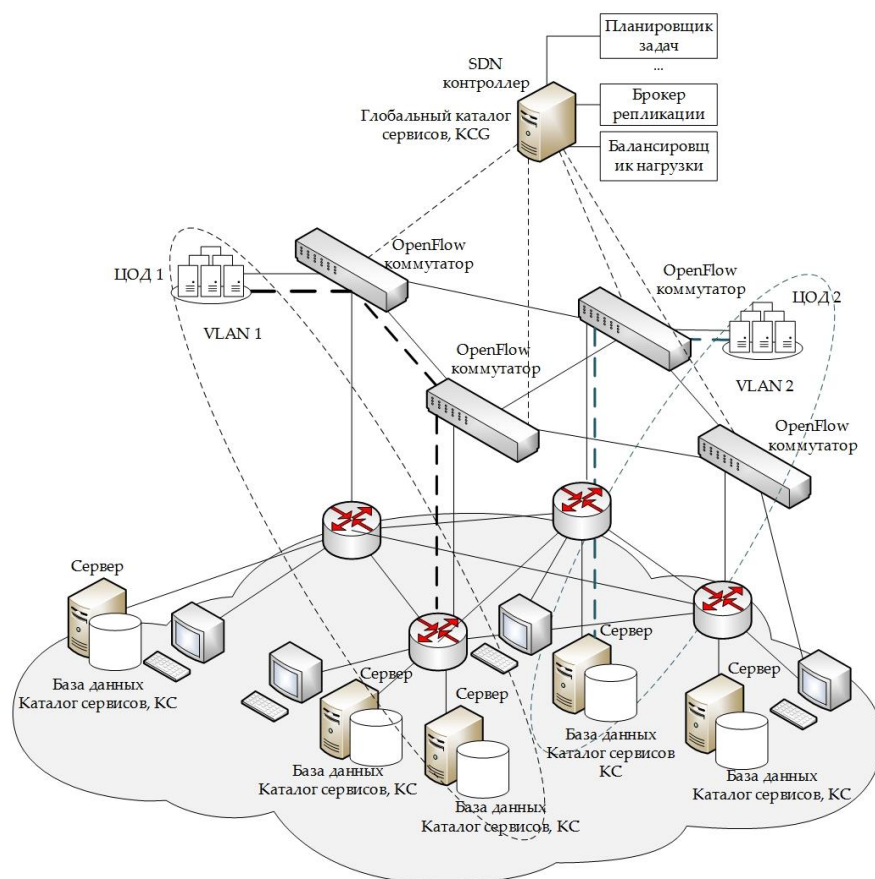


Рис.1. Структурная схема разделения SDN на виртуальные локальные сети

Детализированный сценарий формирования и схема предоставления комплексного сервиса в SDN сети приведены на рис. 2.

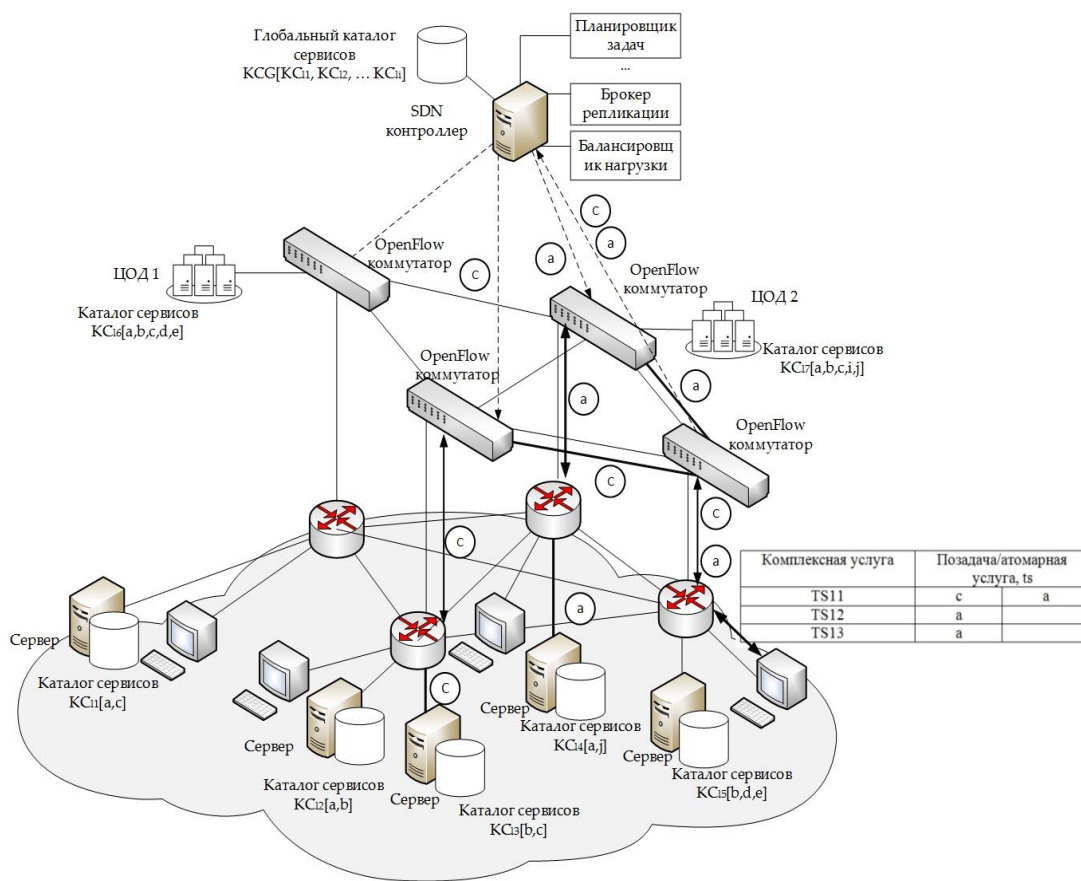
Предположим, что множество потребителей сервиса определено как множество $C = \{c_1, c_2, \dots, c_m\}$, множество предоставляемых сервисов определено как $SC = \{SC_1, SC_2, \dots, SC_n\}$, где SC_i – комплексный сервис, который формируется посредством композиции атомарных сервисов sa_{ki} : $SC_i = \{sa_{1i}, sa_{2i}, \dots, sa_{ki}\}$.

Результующее решение о формировании и предоставлении комплексного сервиса осуществляется на основании информации об атомарных сервисах, входящих в его состав и их месторасположении в сети.

SDN обладает множеством локальных каталогов сервисов (рис. 1). Каждый локальный каталог сервисов $KC_{ij}[sa]$, закреплен за определенной VLAN или LAN, сформированной посредством объединения нескольких OpenFlow коммутаторов [1, 3]. Каждый $KC_{ij}[sa]$ имеет свою область обслуживания, границы которой определяются поставщиком услуг в процессе конфигурирования сети.



а)



б)

Рис. 2. Сценарий формирования (а) и схема предоставления (б) комплексного сервиса в SDN сети

В случае использовании такого подхода информация о каждом i -м сервисе SC_i , которая хранится в локальном каталоге сервисов $KC_{ij}[sa]$, содержащемся в каждой зоне обслуживания, характеризуется множеством следующего вида:

$$KC_{ij}[sa] = \{l(sa), c(sa), t(sa), p(sa), r(sa)\}, \quad (1)$$

где $l(sa)$ – локация атомарного сервиса, $c(sa)$ – стоимость сервиса, $t(sa)$ – время отклика, $p(sa)$ – производительности вычислительного узла, на котором размещен сервис, $r(sa)$ – надежность.

В глобальном каталоге сервисов содержится полная информация о сервисах, предоставляемых в данной сетевой архитектуре. Структура глобального каталога представляет собой упорядоченный массив вида:

$$KCG[sa] = \{KC_{i1}(sa), KC_{i2}(sa), \dots, KC_{ij}(sa)\}. \quad (2)$$

Результирующее значение при формировании комплексного сервиса в SDN имеет функция интеграции атомарных сервисов, которая позволяет определять значение/весовой коэффициент каждого атомарного сервиса, входящего в состав комплексного. Функция интеграции атомарных сервисов имеет следующий вид:

$$IF(SC_i) = K_1(sa_{1i}) + K_2(sa_{2i}) + K_3(sa_{3i}) + \dots + K_k(sa_{ki}), \quad (3)$$

где K_1, K_2, \dots, K_k – весовые коэффициенты атомарных сервисов. Результирующее решение о составе комплексного сервиса определяется исходя из весовых коэффициентов атомарных сервисов.

При формировании комплексного сервиса справедливы следующие выражения:

$$\begin{cases} \forall SC_i, l(SC_i) = \bigcup l(sa_{ki}); \\ p(SC_i) = \min_k p(sa_{ki}); \\ c(SC_i) = \sum_{k=1}^p c(sa_{ki}); \\ r(SC_i) = \min_k r(sa_{ki}), \end{cases} \quad (4)$$

где p – количество атомарных сервисов, входящих в состав комплексного SC_i , $l(sa_{ki})$ – месторасположение i -го атомарного сервиса, входящего в состав комплексного, $c(sa_{ki})$ – стоимость предоставления i -го атомарного сервиса, $p(sa_{ki})$ – производительности вычислительного узла, содержащего i -й атомарный сервис, $r(sa_{ki})$ – надежность.

SDN контроллер формирует состав комплексного сервиса, определяет композицию и месторасположение атомарных сервисов в зависимости от ряда факторов: загруженности сетевых ресурсов, политики обслуживания поставщика услуг, приоритета и объемов предоставляемой услуги и т.д.

Основными требованиями, при формировании комплексного сервиса являются следующие [21]:

- наличие конечного множества атомарных сервисов;
- QoS показатели и стоимость результирующего комплексного сервиса должны соответствовать требованиям к качеству обслуживания, утвержденным в SLA:

$$\begin{cases} QoS(SC_i) = K_1(sa_{1i}) + K_2(sa_{2i}) + \dots + K_k(sa_{ki}) = \sum_{k=1}^n K_k \cdot sa_{ki} \rightarrow \mathbf{max}; \\ QoS(SC_i) \geq QoS_{SLA}(SC_i). \end{cases} \quad (5)$$

$$\begin{cases} C_s(SC_i) = c_1 \cdot sa_{1i} + c_2 \cdot sa_{2i} + \dots + c_k \cdot sa_{ki} = \sum_{k=1}^n c_k \cdot sa_{ki} \rightarrow \mathbf{min}; \\ C_s(SC_i) \leq C_{sSLA}(SC_i), \end{cases} \quad (6)$$

где $QoS(SC_i)$ – результирующее значение качества комплексного сервиса; $QoS_{SLA}(SC_i)$ – качество комплексного сервиса, оговоренное в SLA, c_1, c_2, \dots, c_k – стоимость атомарных сервисов, $C_s(SC_i)$ – результирующее значение стоимости комплексной услуги, $C_{sSLA}(SC_i)$ – значение стоимости, соответствующее SLA.

III. Метод динамической репликации сервисов в программно-конфигурируемых сетях

Принятие решения о необходимости репликации сервисов

Принятие решения о необходимости репликации сервисов осуществляется на основании результатов анализа его популярности и степени деградации QoS с ростом его популярности. Как отмечено в работах [16, 18] значения порога репликации влияет на частоту формирования и распределения копий сервиса.

Выбор значения порога репликации зависит от ряда факторов: объема данных, которые должны быть переданы при репликации, значения ToS (Type of Service) и текущих физических характеристик сети, влияющих на доступность услуги. Тогда пороговое значение репликации может быть представлено как:

$$Th(SC_i) = f(val(SC_i), ToS(SC_i), A(SC_i)), \quad (7)$$

где $val(SC_i)$ – объем данных, переданных по сети в процессе предоставления сервиса, $ToS(SC_i)$ – приоритет сервиса, $A(SC_i)$ – значение доступности сервиса, $A(SC_i)$ носит

переменное значение, зависящее от таких факторов, как текущая популярность сервиса и значение производительности вычислительного узла, на котором расположен сервис.

Текущая популярность сервиса определена интенсивностью запросов на предоставление сервиса за определенный промежуток времени. С целью индикации популярности сервиса в качестве дополнительного атрибута брокера репликации предлагается ввести счетчик запросов на предоставление i -го сервиса с учетом месторасположения источников запроса: $cnt(\mathbf{Req}(SourceIP), KC_{ij}[SC_i])$, где $SourceIP$ – WAN IP-адрес OpenFlow коммутатора той сети, где был сформирован запрос, Req – количество поступивших запросов, $KC_{ij}[SC_i]$ – каталог сервисов, в котором обнаружен запрашиваемый i -й сервис. Таким образом, брокер реплик обладает следующей информацией:

$$BR_{SDN} : [l(SC_i), c(SC_i), t(SC_i), p(SC_i), r(SC_i), CNTS]; \quad \forall SC_i, SC_i \in \{SC\}, \quad (8)$$

где $CNTS$ – множество счетчиков запросов на предоставление i -го сервиса.

Взаимосвязь между порогом репликации и производительностью вычислительного узла, на котором размещен сервис, определяется следующим образом:

$$Th(SC_i) = K_R \cdot p(SC_i), \quad (9)$$

где K_R – коэффициент, определяющий требования к репликации. Если значение $K_R < 1$, то репликация выполняется с целью предотвращения отказов в обслуживании. При $K_R > 1$ основной целью репликации является разгрузка вычислительного узла при значительном увеличении потока заявок.

В случае, если количество запросов на предоставление i -го сервиса, поступивших из одного фрагмента сети, превышает порог репликации:

$$cnt(\mathbf{Req}(SourceIP), KC_{ij}[SC_i]) > Th(SC_i) \quad (10)$$

и приводит к ухудшению качества сервиса, то брокер принимает решение о необходимости запуска процесса репликации.

Определение количества реплик и их расположение в сети

С целью обеспечения гарантированного качества услуг в процессе репликации данные переносятся в тот фрагмент сети, где популярность услуги превышает граничный предел.

Учитывая тот факт, что запрашиваемый пользователем сервис не всегда может находиться в рамках области обслуживания локального каталога сервисов $KC_{ij}[SC_i]$, а также то, что популярность удаленного сервиса может со временем увеличиться, в

распределенных сетях, частности SDN применяется два вида репликации: локальная и удаленная [18, 20].

Необходимость в локальной репликации возникает тогда, когда к определенному сервису в сети резко возрастает количество запросов от потребителей этой сети и это приводит к появлению отказов. Таким образом, создание дополнительной реплики при локальной зоне позволяет повысить доступность и производительность запрашиваемого сервиса, путем перераспределения запросов между репликами.

Необходимость в удаленной репликации возникает в том случае, когда количество запросов из одного участка сети к определенному удаленному ресурсу значительно возрастет. Поэтому, с целью уменьшения трафика передаваемого по сети, а также сокращения времени доступа к ресурсу запускается процесс репликации. Основное преимущество использования удаленной репликации состоит в том, что доступ к сервисам может выполняться локально, без поглощения сетевого трафика и задержек.

Количество реплик i -го сервиса, распределенных между фрагментами сети, определяется следующим образом:

$$n_R(SC_i) = \left\lfloor \frac{F_{R(SourceIP)}(SC_i)}{\sum F_R(SC_i)} \times mSC_i \right\rfloor, \quad (11)$$

где $F_{R(SourceIP)}(SC_i)$ – фактор репликации, соответствующий определенному фрагменту сети, $\sum F_R(SC_i)$ – суммарный фактор репликации, mSC_i – общее количество созданных копий сервиса.

Проверка непротиворечивости реплик

При выполнении синхронной репликации в качестве протокола обеспечения непротиворечивости предложено использовать протокол первичного архивирования [21]. В случае внесения изменений в первичный экземпляр сервиса информация о его обновлении пересылается в локальный каталог сервисов, принадлежащий данной зоне, и глобальный каталог, находящийся на уровне управления. SDN контроллер обеспечивает проверку содержимого глобального каталога и в случае, если обнаружены копии услуги, посредством протокола OpenFlow выполняется обновление обнаруженных копий.

В целом на основании решения приведенных выше задач (2)-(10) может предложен комплексный метод динамической репликации сервисов в SDN (рис. 3).

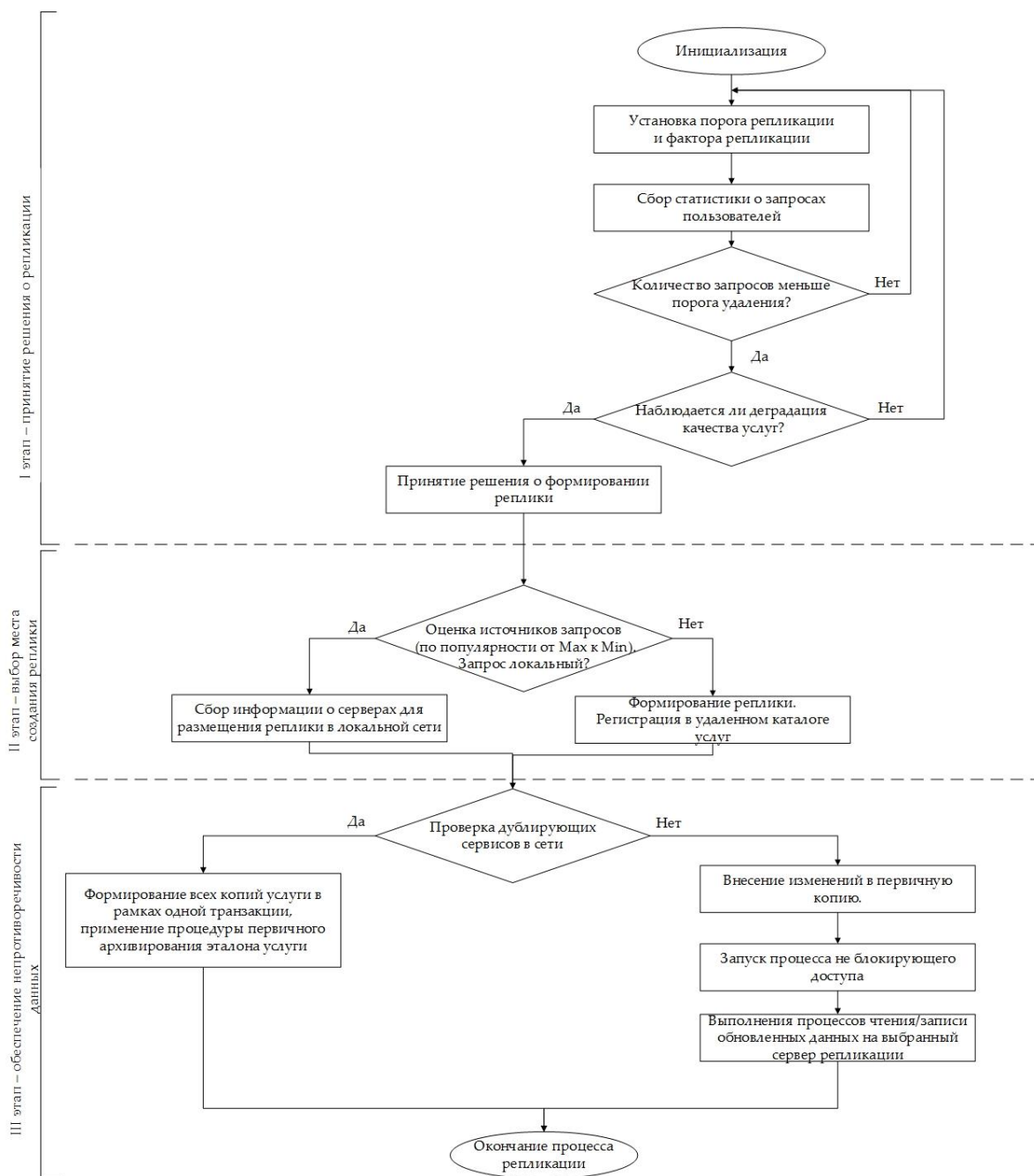


Рис.3. Комплексный метод динамической репликации сервисов

Приведенный на рис. 3 комплексный метод динамической репликации включает выполнение трех этапов: своевременное принятие решения о необходимости формирования реплики сервиса и проведения операции репликации; определение количества и выбор месторасположения реплик; обеспечение непротиворечивости новой реплики с существующими сервисами. Применение данного метода позволяет системе управления гибко адаптировать сетевые ресурсы в соответствии с требованиями пользователей, что способствует оперативному предоставлению сервисов и повышению уровня качества обслуживания в SDN.

Выводы

Применение концепции SDN в крупномасштабных распределенных сетях позволяет значительно повысить качество обслуживания конечных пользователей. Улучшение достигается за счет формирования централизованной системы управления, а также уменьшения нагрузки на сеть передачи данных путем отделения уровня управления от уровня передачи данных. Однако наряду с множеством преимуществ остается ряд нерешенных задач.

Одной из подобных задач является обеспечение требуемого уровня доступности сервисов, эффективным решением которой является применение механизмов репликации. В статье предлагается метод репликации, базирующийся на динамическом принятии ряда решений: выбор момента времени для создания реплик, расчет их количества и определение месторасположения, а также проверки непротиворечивости новой реплики существующим в соответствии с текущим состоянием сети и объемом поступающих запросов. При этом в качестве исходных данных выступали порог репликации, популярность сервиса, производительность вычислительных устройств и локация запросов.

Отличительной особенностью предлагаемого в статье метода является учет основных особенностей функционирования SDN в процессе репликации. Применение предложенной модели формирования и предоставления сервисов, а также динамического метода репликации позволяет своевременно определять причины деградации качества сервисов и устранять их.

Список литературы:

1. SDN architecture [Электронный ресурс] // Open Networking Foundation. – 2016. – 68 p. – Режим доступа: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf.
2. *Feamster N., Rexford J., Zegura E.* The Road to SDN: An Intellectual History of Programmable Networks // ACM Queue. – 2013. – P. 20–40.
3. *Egawa T.* SDN standardization Landscape from ITU-T Study Group // ITU Workshop on SDN, Geneva, Switzerland, 4 June 2013. – 22 p.
4. *Costas N. Atanasova I. K.* OpenFlow and SDN. Technical Report // Technical Report CESGA-2014-001. – 2014. – 64 p.
5. *Issam S.* Mathematical models for analysis Software-Defined Network // Information Technologies & Knowledge. – 2015. – Vol. 9, No.2. – P.111-123.
6. *Lawrence S.* Software Engineering: Theory and Practice. – Upper Saddle River, NJ: Prentice Hall, 2001. – 630 p.
7. *Infonetics Research, Inc.* SDN and NFV Strategies: Global Service Provider Survey [Электронный ресурс]. – 2014 – 39 с. – Режим доступа: <http://alu.us.neolane.net/res/img/286758382c7e061c52883e873cee02e6.pdf>.
8. *Akyildiz I. F.* A roadmap for traffic engineering in SDN-OpenFlow networks // Computer Networks. – 2014. – Vol. 71, No.4. – P. 1-30.

9. *Stantchev V.* Effects of Replication on Web Service Performance in WebSphere // ICSI Technical Report TR-08-003. – 2008. – 72 p.
10. *Yousefi A.* Request Replication: An Alternative to QoS Aware Service Selection // Proceedings of the 2011 IEEE International Conference on Service-Oriented Computing and Applications, SOCA. – 2011. – P. 1-14
11. *Khan S.U., Ahmad I.* Comparison and analysis of ten static heuristics-based internet data replication techniques // Parallel Distrib. Comput. – 2008. – 68(2). – P.113–136.
12. *Malik S. U., Madani S. A., Khan S. U.* Performance analysis of data intensive cloud systems based on data management and replication: a survey // Distributed and Parallel Databases. – 2016. –Vol.34, No.2. – P. 179-215.
13. *Chang R. S. Chang H. P.* A dynamic data replication strategy using access-weights in data grids // Journal of Supercomputing, – 2008. – Vol.45, No. 3. – P. 277-295.
14. *Kia H.S. Khan S.U.* Server replication in multicast networks // 10th IEEE International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan. – 2012. – P. 337–341
15. *Ikeda T., Ohara M., Fukumoto S.* A distributed data replication protocol for file versioning with optimal node assignments // Proceedings of IEEE International Pacific Rim International Symposium on Dependable Computing 2010. – 2010. – P.117–125
16. *Wei Q., Veeravalli B., Gong B. and all* CDRM: A cost-effective dynamic replication management scheme for cloud storage cluster // IEEE International Conference on Cluster Computing 2010. – 2010. – P. 188–197.
17. *Bonvin N., Papaioannou T.G., Aberer K.* A self-organized, fault tolerant and scalable replication scheme for cloud storage // Proceedings of the Symposium on Cloud Computing, Indianapolis, USA. – 2010. – P. 205–216.
18. *Khan S., Ahmad I.* Heuristic-based replication schemas for fast information retrieval over the internet // Proceedings of 17th International Conference on Parallel and Distributed Computing Systems. – 2004. – P. 278–283.
19. OpenFlow Switch Consortium and Others. OpenFlow Switch Specification Version 1.2.0 [Электронный ресурс] // OpenFlow Switch Consortium and Others. – 2011. – Режим доступа: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/>.
20. *Yang Y., Tang S., Xu Y. and all.* An Approach to QoS-Aware Service Selection in Dynamic Web Service Composition // 3rd IEEE International Conference on Networking and Services (ICNS). – 2007. – P.18–23.
21. *Дейт К. Дж.* Введение в системы баз данных, 8-е издание. – М.: Издательский дом "Вильямс", 2005. – 1328 с.