На правах рукописи

Красулин Георгий Алексеевич

ИССЛЕДОВАНИЕ МЕТОДОВ ОБЕСПЕЧЕНИЯ НАДЕЖНОСТИ ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ (SDN)

Направление подготовки 11.04.02
«Инфокоммуникационные технологии и системы связи»
профиль: Многоканальные телекоммуникационные системы
программа академической магистратуры

АВТОРЕФЕРАТ
магистерской диссертации
на соискание квалификации (степени) магистра

Екатеринбург 2020

Работа выполнена в федеральном государственном образовательномбюджетном учреждении высшего профессионального образования «Уральскийтехнический институт связи и информатики (филиал) Сибирскогогосударственного университета телекоммуникаций и информатики» (г.Екатеринбург).

Научный руководитель
к.т.н доцент Н.В. Будылдина

Рецензент:
к.т.н. Д.В. Кусайкин

Защита состоится «1» декабря 2020 г. В 9.00 часов в Федеральном государственномбюджетном образовательном учреждении высшегообразования «Сибирскийгосударственный университет телекоммуникаций и информатики» Уральскийтехнический институт связи и информатики(филиал) в г. Екатеринбурге(УрТИСИСибГУТИ), г. Екатеринбург, ул. Репина, д. 15.

Секретарь Государственной аттестационной комиссии О.А. Шумилова

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования и степень ее разработанности.

Актуальность темы исследования методов обеспечения надежности SDNобусловлена тем фактом, что высокий темп развития сервисов, масштабов их охвата, а так же рост количества и вариативности контента привели к изменению концепции организации вычислений – место устаревшей клиент-серверной архитектуры заняли центры обработки данных (ЦОД) и облака. В то же время, сетями хранения данных стали файловые системы и базы данных. Рост показателей нагрузки усложнился возникающими проблемами в управлении сетями, причем на фоне повышения требований к безопасности и надежности. Еще одной немаловажной проблемой является постоянное обновление программного обеспечения и аппаратных средств, что влечет за собой приобретение дополнительного оборудования и привлечение специалистов, которые будут заниматься настройкой и поддержкой этого оборудования. Построение сети на базе постоянно усложняющихся устройств, подразумевает использование большого количества протоколов маршрутизации. Все они отличаются своими принципами работы, типами и алгоритмами. Как результат – невозможность провайдера вводить новые сервисы, вдобавок, производители сетевого оборудования не успевают модернизировать свои устройства для удовлетворения требований заказчиков. Одним из возможных вариантов решения большинства вышеперечисленных проблем может стать применение технологии SDN, что позволит изменить представление и существенно повлиять на подход к построению сетей связи. И чтобы вся эта система работала и выдавала максимально качественный результат, необходимо проанализировать и сравнить методы обеспечения надежности и безопасности сетей SDN и выявить лучший и качественный метод.

Объект исследования – программно-конфигурируемые сети.

Предмет исследования – обеспечение надежности в программно-конфигурируемых сетях.

Целью работы является исследование методов обеспечения надежности сетей SDN.

Для достижения обозначенной цели необходимо решить следующие задачи:

1) исследовать современные проблемы надежности и безопасности сетей SDN;

2) проанализировать современные проблемы повышения надежности сетей SDN;

3) сравнить методы повышения надежности с другими предложенными методами, сделать вывод почему метод, который был выбран надежнее и безопаснее остальных методов;

4) произвести расчет показателя надежности выбранного метода и оценить выбранный метод после получения численных значений.

Научная новизна заключается в определении более надежного метода передачи данных в программно-конфигурируемых сетях.

Практическая значимость заключается в определении наилучшего алгоритма обеспечения надежности при полученных расчетах и сравнении с ранее предложенными методами с учетом параметров времени восстановления, время задержки и скорость передачи.

Теоретическая значимость заключается в выборе методов обеспечения надежности в программно-конфигурируемых сетях.

Методология исследования. При выполнении диссертационной магистерской работы были изучены несколько методов обеспечения надежности, выделены достоинства и недостатки каждого метода. Проведено сравнение всех методов друг с другом и с более перспективным методом RDSDN, произведены математические расчеты и все полученные результаты сведены в общие графики и таблицы для наглядного сравнения результатов исследования.

Положения, выносимые на защиту:

1) метод обеспечения надежности RDSDN;

2) методика расчета средней времени задержки с коэффициентом потерь, времени восстановления и пропускной способности;

3) алгоритм определения надежности с использованием средней времени задержки, времени восстановления и пропускной способности с использованием пакета MathLab.

Степень достоверности результатов исследования подтверждается корректностью постановки задач, применение математических моделей, непротиворечивостью результатов и выводов. Моделированием, а также сравнение полученных результатов с известными результатами.

Апробация результатов. Основные результаты диссертации были получены и использованы в рамках проекта Научно-производственное предприятие «Медпромдеталь». Материалы магистерской диссертации были изложены в статье. Центр научной мысли (г. Таганрог). Результаты диссертации обсуждались на всероссийской научно-практической конференции «Информационные технологии и когнитивная электросвязь», (УрТИСИСибГУТИ, Екатеринбург, 2020), в Студенческий вестник №38(136), 19 октября 2020г.

Структура и объем диссертации. Диссертационная работа включает введение, 3 главы, заключение, библиографический список из 33 наименований.

Объем диссертации 89 листов, включены 22 рисунков.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы, сформулированы цель и задачи исследования, определены научная новизна и практическая значимость работы, изложены основные положения, выносимые на защиту.

В первой **главе «Анализ публикаций по теме исследования»** Произведен анализ публикаций по теме исследования. Теоретические исследования в области сетей SDN были рассмотрены следующими авторами: Захаров, А. А., Коломеец А. Е., Попов, М. М., Омелич П.П., Сурков Л. В., Арестов А.А., Смелянский Р.Л., Деарт, В.Ю., Абаева, Б.К, Лапонина, О. Р., Серов, А.А., Лапонина, О.Р., В. А. Сухомлин., Ижванов Ю.Л., Олифер, Н.В., Горелик С.Л.

Таким образом, технология SDN может существенно снизить уровень нагрузки на сеть, упростить общую концепцию построения сетей, а, следовательно, и процедуру администрирования сетей. Вместе с этим возникают вопросы по обеспечению безопасности новой технологии передачи данных, так как для новых технологий встречаются новые проблемы, предусмотреть которые невозможно при их разработке.

Теоретический анализ литературы показывает, что, исследование SDN сетей актуальная и необходимая задача сегодня, особенно касательно безопасности SDN.

Во второй главе «**Теоретические исследования в области применения технологии SDN**»диссертации были рассмотрены программно-конфигурируемые сети. Было показано на каких трех уровнях сети SDNработают. На рисунке 1 показаны уровни работы сетей SDN.

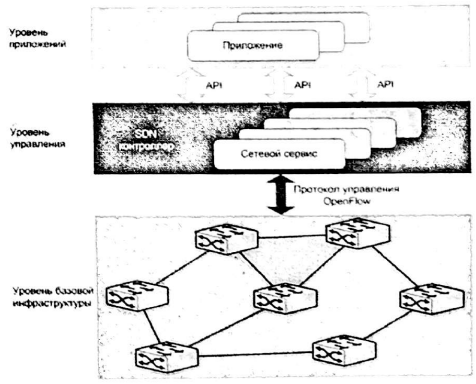


Рисунок 1 – Уровни SDN

Вся это система работает на трех уровнях:

- уровень инфраструктуры;
- уровень управления;
- уровень приложений.

Уровень инфраструктуры – на этом уровне представлены сетевое оборудование.

Уровень управления – контроллер управления или SDN-контроллер, который управляет всем сетевым оборудованием, генерирует и отправляет коммутаторам и маршрутизаторам правила передачи трафика.

Уровень приложений – взаимодействует с SDN-контроллером через программный протокол (ApplicationProgramInterface – API) для сбора, анализа, развертывания и управление сетевой инфраструктурой на уровне приложений.

Были рассмотрены методы обеспечения надежности, каждый метод подробно описан и приведены все достоинства и недостатки каждого метода. После сравнения всех методов:

- RDSDN;
- FCF-M;
-NSNC;
- PPF;
- FLCF;
- Survivor.

Был выявлен самый надежный метод обеспечения надежности, это метод RDSDN. RDSDN – единственный метод, который учитывает все упомянутые показатели при вычислении надежности и выборе хотя бы контроллера для выполнения процесса восстановления и переназначения отказавших коммутаторов в глобальном масштабе, что приводит к улучшению коммутатора уменьшению задержки контроллера.

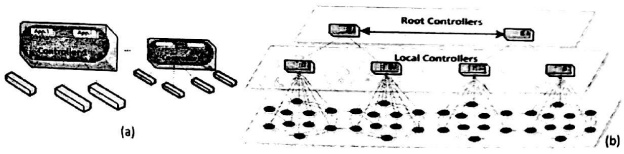


Рисунок 2 – Вид распределенных контроллеров

Архитектура распределенного контроллера предпочтительнее, когда важна надежность. Тем не менее топология плоскости управления и уровень распределения сетевых элементов по-прежнему представляют ряд альтернатив с различными характеристиками. Очевидно, что распределение элементов сети между контроллерами может снизить влияние неизбежных физических отказов на управляющие сети и повысить общую надежность.

Основное предположение этой работы состоит в том, что большая сеть может быть разделена на более мелкие подсети для лучшего контроля. Компоненты каждой подсети взаимосвязаны и работают независимо. Таким образом, вся сеть становится менее подверженной отказам узлов передачи или контроллеров. Основная идея предложения RDSDN состоит в том, что главный контроллер управляет каждой подсетью, и в то же время каждая подсеть имеет один или несколько контроллеров других подсетей в качестве подчиненных контроллеров. Главные контроллеры рассчитывают степень надежности своих подсети в зависимости от количества узлов, подключенных к контроллеру, а также количества подключенных и взаимосвязанных каналов и их коэффициентов потерь.

В третьей главе **«Исследование методов обеспечения надежности»** определены показатели надежности каждого из предложенных методов и проведено математическое моделирование полученных результатов.

И чтобы понять какой метод более надежный, необходимо провести расчеты показателя надежности каждого метода по нескольким критериям:

- время задержки;
- время восстановления после сбоя;
- пропускная способность.

Каждая подсеть моделируется вероятностным графом G (N, D, Q l), где N – набор узлов (рисунок 2). Каждый коммутатор или контроллер является узлом на графе, а физическая связь между двумя узлами представлена ребром. Зная N, мы затем вычисляем вектор D размера N, состоящий из степеней каждого узла, и матрицу Q1 с размерами N×N, представляющую скорость потери связи между любыми двумя узлами. Предполагается, что коэффициент потери канала является экспоненциальной функцией коэффициента потери пакетов, то есть процента от количества потерянных пакетов в зависимости от количества пакетов, отправленных по каналу. Чтобы создать матрицу Ql, сначала каждый контроллер генерирует матрицу размерности N на N (рисунок 3). Если существует прямая ссылка от узла i к узлу j, вероятность потери связи для этой связи записывается в соответствующей записи матрицы. В противном случае значение будет установлено на 0, за исключением случая, когда i = j, где значение установлено на 1.

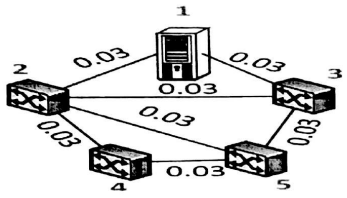


Рисунок 2 – Подсеть с графом G (5, D, Q l)

$$Q1=\begin{vmatrix} 1 & 0.03 & 0.03 & 0 & 0 \\ 0.03 & 1 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.03 & 1 & 0 & 0.03 \\ 0 & 0.03 & 0 & 1 & 0.03 \\ 0 & 0.03 & 0.03 & 0.03 & 1 \end{vmatrix}$$

Рисунок 3 – Матрица Q1

Рассматривая топологию подсети на рисунке 2 с графом G (5, D, Q 1) и коэффициентом потерь в канале 0,03, вектор степени D и матрица Q 1 будут выражены следующим образом.

Надежность системы, представленной формулой R(G) можно определить как вероятность того, что каждая пара узлов может отправлять и получать данные. Каждый связный граф представляет собой остовое дерево с N узлами и N −1 ребром. Таким образом, надежность этого связного графа G (N, N −1, q) выражается уравнением (1). q – вероятность отказа каналов, которая одинакова для всех каналов, а вероятность срабатывания всех каналов равна 1-q.

$$R\left(G_{(N,N-1,q)}\right) = (1-q)^{N-1} \tag{1}$$

Набор минимальных разрезов применяется как последовательность звеньев, удаление которых разъединит G. В уравнении (2) C_j – это событие, вызывающее сбой всех связей в j-м минимальном наборе отсечений и $\bar{C}_j = 1 - C_j$.

$$\begin{aligned} 1-R(G) &= \Pr(C1) + \Pr(C2 \cap \bar{C}1) + \ldots \\ &+ \Pr(CM \cap \bar{C}1 \cap \bar{C}2 \cap \ldots \cap \bar{C}M-1 \\ &\geq \Pr(F1) + \Pr(F2 \cap \bar{F}1) + \ldots \\ &+ \Pr(FN \cap \bar{F}1 \cap \bar{F}2 \cap \ldots \cap \bar{F}N-1 \end{aligned} \tag{2}$$

Алгоритм 1 Расчет v_i в подсети.

Вход: граф подсети G (N, D, Q_1) с возможными взаимосвязанными связями между подсетью и соседними, которые применяются в Q_l, и узлом i в подсети.

Выход: доля вероятности потери узла i.

Решение:

1: $v_i = 1$;
2: For ve: 1 to N do
3: IF Ql (i, ve)! = 0, then v_i = Ql (i, ve) *v_i;
4: end IF
5: Endfor

Алгоритм 2 Вычисление v_j в подсети.

Вход: граф подсети G (N, D, Ql) с возможными взаимосвязанными граничными связями между подсетью и соседними, которые применяются в Ql, и узлом j в подсети.

Выход: доля вероятности потери узла j.

Процедура:

1: $v_j = 1$;
2: For ve: 1 to N do
3: IF Ql (j, ve)! = 0, then v_j = Ql (j, ve) *v_j;
4: endIF
5: Endfor

Алгоритм 3 Вычисление v_j 2 в подсети.

Вход: граф подсети G (N, D, Ql) и возможные взаимосвязанные граничные ссылки между подсетью и соседними, которые применяются в Ql, а также узлы i и j в подсети при условии, что ссылка из узелі к узлу j не существует

Выходные данные: доля вероятности потери узла j, когда ссылка от узла i к узлу j не существует.

Процедура:

1: $v_{j2} = 1$;
2: For ve: 1 to N do
3: IF Ql (j, ve)! = 0, then v_{j2} = Ql (j, ve) *v_{j2};
4: end IF
5: End for
6. IF Ql (j, i)! = 0, then $v_{j2} = v_{j2}$ / Ql (j, i);
7: endIF

В уравнении (1), предполагается, что q одинаково в каждом из звеньев. Однако в SDN показатели потери связи могут отличаться. Вероятность отказа канала от узла i к узлу j будет Ql (i, j) в соответствии с данной подсетью и матрицей Ql. Если ссылка (i.j) существует в G, его направленные узлы, i и j, имеют долю вероятности потери v_i и v_j, соответственно. Если ссылка от узла i к узлу j не существует, доля вероятности потери будет vj 2. Таким образом, вероятность того, что ссылки, подключенные к узлу i и узлу j, не откажутся ($\bar{F}_i \cap \bar{F}_j$) выражается формулой (3), и вероятность того, что все ссылки, подключенные к узлу i, выйдут из строя при условии, что ссылки, подключенные к узлу j, не откажут ($F_i \cap \bar{F}_j$), выражается уравнением (4). V_i, v_j и v_{j2} вычисляются с помощью алгоритмов 1-3.

Вероятность события ($\bar{F}_i \cap \bar{F}_j$).

$$v_i x v_i. \text{ if link } (i.j) \in G. \tag{3}$$

И вероятность события $F_i \cap \bar{F}_j$ =

$$\begin{cases} v_i(1-v_i). if link(i.j) \epsilon\, G \\ v_i(1-v_{i2}). if link(i.j) \epsilon G \end{cases} \tag{4}$$

Рассматривая исходный узел i и возможные узлы j и k, событие $(F_i \cap \overline{F_j} \cap \overline{F_k})$ исследуется в двух условиях: когда количество связей, подключенных к узлу i (d_i), больше, чем или равно двум, или когда меньше двух. Эти условия выражены в уравнениях. (5) и (6) соответственно.

Вероятность события $(F_i \cap \overline{F_j} \cap \overline{F_k} \mid d_i \geq 2)$. с учетом вероятностей потери связи среди них:

$$\Pr(F_i \cap F_j \cap F_k) =$$
$$v_i(1-v_j)(1-v_k).\ \text{if link (i. j)} \in G \text{ and link (i. k)} \in G$$
$$v_i(1-v_{j2})(1-v_k).\ \text{if link (i. j)} \in G \text{ and link (i. k)} \in G$$
$$v_i(1-v_j)(1-v_{k2}).\ \text{if link (i. j)} \in G \text{ and link (i. k)} \in G$$
$$v_i(1-v_{j2})(1-v_{k2}).\ \text{if link (i. j)} \in G \text{ and link (i. k)} \in G. \quad (5)$$

Вероятность события $(F_i \cap \overline{F_j} \cap \overline{F_k} \mid d_i < 2)$ с учетом вероятностей потери канала среди них составляет:

$$\Pr(F_i \cap F_j \cap F_k) =$$
$$v_i(1-v_j)(1-v_k).\ \text{if link (i. j)} \in G \text{ and link (i. k)} \in G$$
$$v_i(1-v_{j2})(1-v_k).\ \text{if link (i. j)} \in G \text{ and link (i. k)} \in G$$
$$v_i(1-v_j)(1-v_{k2}).\ \text{if link (i. j)} \in G \text{ and link (i. k)} \in G. \quad (6)$$

Затем,

$$\Pr(F_i \cap \overline{F_j} \cap \overline{F_k}) \geq \begin{cases} v_i(1-v_{j2})(1-v_{k2}).\ \text{if } d_i \geq 2 \\ v_i(1-v_j)(1-v_{k2}).\ \text{if } d_i < 2 \end{cases}. \quad (7)$$

Так,

$$1 - R(G) \geq \sum_{i=1}^{N} v_i \prod_{j=1}^{m_i}(1-v_{j2}) \prod_{j=m_j+1}^{j-1}(1-v_j), \quad (8)$$

$m_i = \min(d_i.i\text{-}1).i=1.2 \ldots N$

Верхняя граница R(G) выражается формулой (9). v_i, v_j и v_{j2} описываются с помощью алгоритмов 1, 2 и 3 соответственно. Поскольку этот алгоритм учитывает топологию подсети, если между двумя подсетями существует какое-либо взаимосвязанное соединение, их вероятность отказа также может быть принята во внимание.

$$R(G) = 1 - \left(\sum_{i=1}^{N} v_i \prod_{j=1}^{m_i}(1-v_{j2}) \prod_{j=m_j+1}^{i-1}(1-v_j) \right), \quad (9)$$

где $m_i = \min(d_i.i\text{-}1).i=1.2 \ldots N$.

Чтобы подтвердить подлинность уравнения (9) и оценки надежности, некоторые образцы и численные результаты представлены в следующем разделе.

Рассмотрим алгоритм координатора. Алгоритм 4 в этом разделе описывает метод поиска наиболее надежного контроллера среди всех распределенных контроллеров. Этот контроллер будет координатором всей сети. Этот подход описан ниже: Каждый главный контроллер рассчитал и сохранил свой коэффициент надежности на основе своей подключенной топологии плоскости данных и скоростей потерь подключенных и взаимосвязанных каналов с помощью уравнения. (9) (строки 1–3 алгоритма 4). После этого каждый главный контроллер отправит уровень надежности через недавно разработанный интерфейс восточно-западного направления и Rest API следующим образом:

- создайте новый пакет, содержащий его уровень надежности и имя в качестве полезной нагрузки;
- выберите крайний переключатель, ближайший к краевому переключателю другого контроллера;
- поместите IP-адрес выбранного пограничного коммутатора в качестве адреса источника и IP-адрес пограничного коммутатора, подключенного к другому контроллеру, в качестве адреса назначения и кэшируйте адреса для дальнейших попыток (строки 4-7 алгоритма 4);
- отправить пакет через новое сообщение Packet-Out (строка 10 алгоритма 4).

Алгоритм 4 Алгоритм поиска координатора.

Вход: широкая сеть, разделенная на (от I = 1 до S) подсети, каждая из которых представлена $G_I = (N_I, D_I, Q_I)$ и управляется главным контроллером C_I.

Выход: Контроллер числа и координатора

Процедура:

1: ForI: 1 toSdo
2: C_I Computes R_I through G_I, D_I and Q_I using
3: C_I stores R_I in its cache
4: For J: 1 to S and (J! = I) do
5: C_I Chooses an edge switch (Source switch) in its subnetwork which is nearest to the edge switch of controller C_J (Destination Switch)
6: C_I caches the address of Source and Destination switches.
7: C_I Creates a Packet containing R_I and I as payload and puts the IP addresses of the selected switches as Source and Destination IP addresses.
8: IF (I = = number) then C_I adds "I am your Coordinator" to the payload.
9: End IF
10: C_I Sends the packet to C_J through a new Packet-out message.
11: End For
12: End For
13: For I: 1 to S do
14: Each edge switch Switch I which receives the packet gives that to its controller C_I as a new Packet-In message
15: End for

16: For I: 1 to S do
17: For J: 1 to S and J! = I do
18: C_I caches the received R_J from C_J along with J in a descending sequence
19: End For
20: IF R_I is greatest comparing to the elements in the cache then
21: Coordinator = C_I and number = I.
22: End IF
23: EndFor

Когда коммутатор, подключенный к другому контроллеру, получает пакет, онотправляет его своему контроллеру как новое сообщение о пакете (строка 14 алгоритма 4).

Каждый главный контроллер отсортирует полученные показатели надежности вместе с ее значением надежности. Таким образом, в качестве координатора будет выбран контроллер с наивысшей степенью надежности (строки 20 и 21 алгоритма 4). Выбранный контроллер объявит об этом, отправив новый пакет, содержащий утверждение «Я ваш координатор», всем остальным контроллерам (строка 8 алгоритма 4). Чтобы уменьшить нагрузку, это объявление будет сделано в том же пакете в течение следующего периода отправки пакетов уровня надежности. Если для каждого контроллера будет больше 3f+1 реплик, этот метод может быть полезен не только для предотвращения сбоев, но и для предотвращения византийских ошибок. Процесс отправки показателей надежности выполняется периодически (каждые двадцать секунд с разницей в пять миллисекунд для каждого контроллера). Таким образом, если координатор выходит из строя или показатели надежности значительно изменяются, новый контроллер может быть выбран путем выбора с помощью такого алгоритма, как алгоритм Bully.

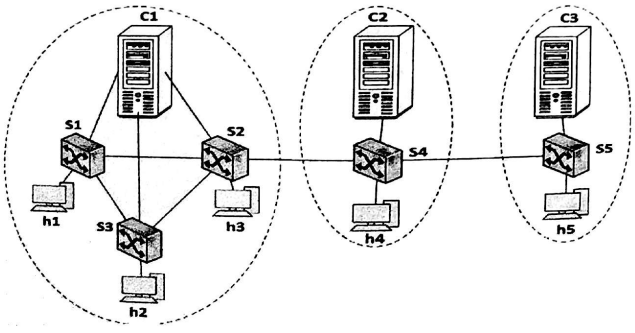


Рисунок 3 – Топология, содержащая три контроллера SDN с разными топологиями подсетей и разной скоростью потери каналов

Данная схема содержит три контроллера SDN с IP-адресами C1 = 192.168.56.101, C2 = 192.168.56.102 и C3 = 192.168.56.103, управляющих разными топологиями подсетей с двумя разными вероятностями потери канала:

Ситуация A: коэффициент потери связи 0,01 во всех подключенных и взаимосвязанных каналах.

Ситуация B: коэффициент потери связи 0,01, 0,02 и 0,05 в C1, C2 и C3 соответственно. Коэффициент потери соединенных каналов составляет 0,01.

Для экспериментальной оценки было реализована топология сети, показанная на рисунке 3, с помощью инструмента эмуляции сети Mininet. Три контроллера были смоделированы с помощью Floodlight. Floodlight – это контроллер OpenFlow на базе Java с открытым исходным кодом, который используется большим сообществом разработчиков и исследователей.

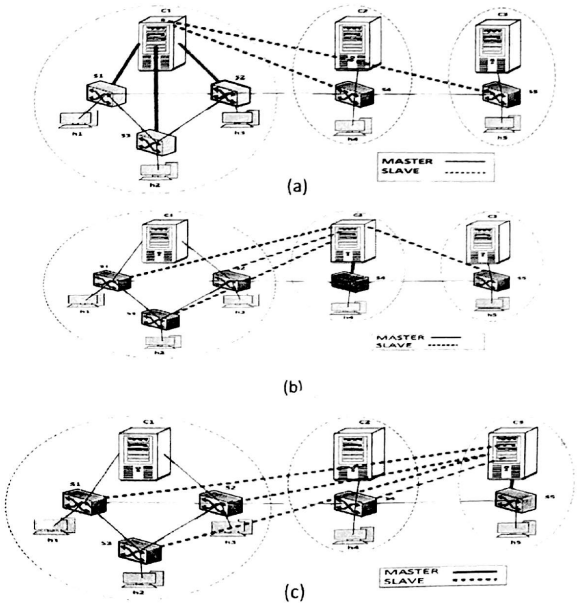


Рисунок 4 – Метод конфигурации Master / Slave для повышения отказоустойчивости контроллеров C1 – C3

Чтобы защитить контроллеры в случае сбоев, на этапе конфигурации реализован метод Master / Slave. В каждой сетевой секции SDN-контроллер секции устанавливает логические отношения (идентификатор пути) с коммутаторами, в которых он играет роль Master. Это означает, что коммутаторы будут обращаться к этому контроллеру SDN для любых действий и принимать от него конфигурации потока. Другие контроллеры всей сети также будут подключаться к коммутаторам с ролью ведомого устройства, то есть существуют контроллеры ведомого устройства для коммутаторов с доступом только для чтения к ним, но не отправляют и не принимают асинхронные сообщения коммутатора и переключатели не получают от них команд.

Алгоритм поиска координатора был полностью реализован в Floodlight для архитектуры эталонной сети, показанной на рисунке 3, с тремя контроллерами. На основе предложенной топологии и алгоритма – 4 C1 затем станет координатором всей сети, который будет хранить в своем кэше необходимую информацию о работе подсети, включая показатели надежности.

Таблица 1 – Показатели надежности трех контроллеров

Название контроллера	Уровень надежности	
	На основе коэффициента потери канала в ситуации A	На основе коэффициента потери канала в ситуации B
C1	0.9898	0.9898
C2	0.9703	0.9605
C3	0.9801	0.9405

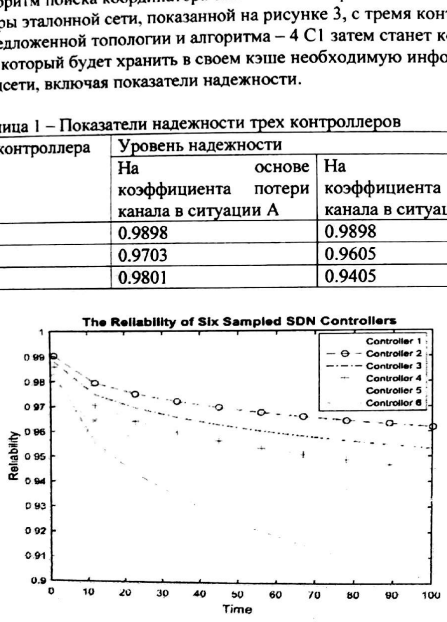
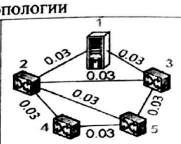
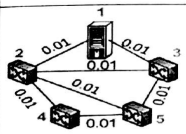
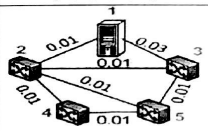
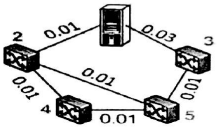
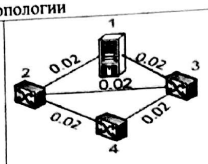
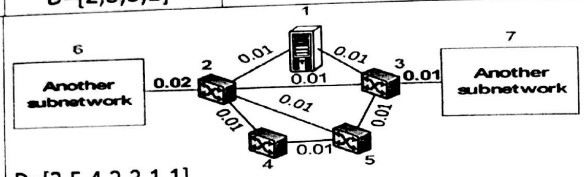


Рисунок 5–Графики надежности шести контроллеров с различными топологиями уровня данных и коэффициентами потерь подключенных и взаимосвязанных каналов

Таблица 2 – Сравнение показателей надежности контроллеров SDN с учетом различных топологий плоскости данных и коэффициентов потери каналов

Вид топологии		Q_i
C1	 D=[2,4,3,2,3]	1 0.03 0.03 0 0 0.03 1 0.03 0.03 0.03 0.03 0.03 1 0 0.03 0 0.03 0 1 0.03 0 0.03 0.03 0.03 1
C2	 D=[2,4,3,2,3]	1 0.01 0.01 0 0 0.01 1 0.01 0.01 0.01 0.01 0.01 1 0 0.01 0 0.01 0 1 0.01 0 0.0 0.01 0.01 1
C3	 D=[2,4,3,2,3]	1 0.01 0.03 0 0 0.01 1 0.01 0.01 0.01 0.03 0.01 1 0 0.01 0 0.01 0 1 0.01 0 0.01 0.01 0.01 1
C4	 D=[2,3,2,2,3]	1 0.01 0.03 0 0 0.01 1 0 0.01 0.01 0.03 0 1 0 0.01 0 0.01 0 1 0.01 0 0.01 0.01 0.01 1

Продолжение таблицы 2

Вид топологии	Q_1
C5  D=[2,3,3,2]	$\begin{vmatrix} 1 & 0.02 & 0.02 & 0 \\ 0.02 & 1 & 0.02 & 0.02 \\ 0.02 & 0.02 & 1 & 0.02 \\ 0 & 0.02 & 0.02 & 1 \end{vmatrix}$
C6  D=[2,5,4,2,3,1,1]	$\begin{vmatrix} 1 & 0.01 & 0.01 & 0 & 0 & 0 & 0 \\ 0.01 & 1 & 0.01 & 0.01 & 0.01 & 0.02 & 0 \\ 0.01 & 0.01 & 1 & 0 & 0.01 & 0 & 0.01 \\ 0 & 0.01 & 0 & 1 & 0.01 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.01 & 1 & 0 & 0 \\ 0 & 0.02 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 1 \end{vmatrix}$
уровень надежности	$R_1(G)=0.9981$
	$R_2(G)=0.9998$
	$R_3(G)=0.9996$
	$R_4(G)=0.9993$
	$R_5(G)=0.9992$
	$R_6(G)=0.9700$

Примечательно, что QoS не является основной целью предлагаемого нами метода RDSDN. Но, чтобы поддерживать QoS в RDSDN, каждый контроллер может отправлять политику QoS для своей подсети координатору или сохранять ее в распределенной файловой системе. На этапе обнаружения и восстановления, когда координатор обнаруживает аварийный отказ контроллера, он выбирает новый контроллер, которому будет принадлежать подсеть отказавшего контроллера. Для выполнения QoS координатор также может отправлять политики QoS новому. Таким образом, этот новый контроллер может

устанавливать политики для добавленной подсети и иметь свои собственные политики QoS для своей собственной подсети.

Если обнаружен отказ более чем одного контроллера, приоритет на этом уровне зависит от расстояния. Основываясь на показателях надежности, координатор имеет упорядоченный список контроллеров и имеет общее представление о сети на основе полученных матриц Q 1. Таким образом, может решить, какая из отказавших подсетей ближе всего к первому члену списка, и сначала выполнить процесс восстановления для этой подсети, а затем для остальных. С помощью этого метода восстановление осуществляется с учетом таких показателей, как надежность и расстояние. Как показано, статус роли C1 для S4 и S5 был изменен на Master.

На этапе оценки надежности для представления вероятности времени отказа обычно применяются кумулятивные функции распределения, такие как экспоненциальное распределение Вейбулла. В данном примере распределение Вейбулла применяется для моделирования надежности, поскольку это гибкая модель распределения с двумя параметрами масштаба и формы, α и β, соответственно. Время до отказа SDN-контроллера C, содержащего его подсеть G, может быть определено распределением Вейбулла, как в формуле (10). Распределение Вейбулла для контроллера SDN i содержащая свою подсеть:

$$F_c(t) = 1 - e^{-\left(\frac{t}{\alpha}\right)^{\beta}} = 1 - e^{-\left(\frac{t}{A(c)}\right)^{\beta}}, \tag{10}$$

$$A(c) = \frac{\gamma}{1 - R_c(G).} \tag{11}$$

Надежность каждого контроллера (C) обеспечивается с помощью уравнения:

$$R_c(t) = 1 - F_c(t). \tag{12}$$

где $R_c(t)$ s различных топологий и коэффициенты потерь в канале, упомянутые в таблице 2, получены через MATLAB версии R2015b с использованием уравнения (12) рисунок 6. Примечательно, что значения надежности были достигнуты при $\gamma = 1000$ и $\beta = 0.3$.

Из-за обработки большого объема запросов, чем больше количество подключенных компонентов, тем выше частота отказов и, следовательно, более низкая надежность, и наоборот, только несколько компонентов могут привести к уменьшению покрытия неисправностей. Напротив, чем больше количество подключенных каналов к контроллеру или плоскости данных, тем выше надежность, как и в таблице. 2, подсеть 3 по сравнению с подсетью 4. Как показано в таблице 2 (подсеть 2), надежность контроллера может увеличиваться при уменьшении скорости потери связи.

После вычисления полученные данные были сведены в MathLab. Результаты моделирования представлены на рисунке 6.

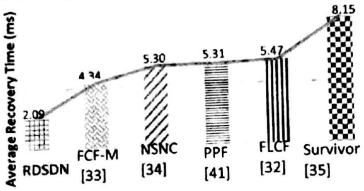


Рисунок 6 – Среднее время задержки переключения на контроллер или сравнение времени восстановления предлагаемого метода RDSDN и других методов

Было проверено поведение пакетов в периоды сбоя и восстановления. 100 пакетов были отправлены из H1 в H5 в шести упомянутых условиях, во время которых контроллер C2 вышел из строя и был быстро обнаружен Координатором, и на основе связанных условий фаза восстановления была завершена. Средняя задержка и потеря пакетов во время сбоя и восстановления были измерены,процесс восстановления не оказывает значительного влияния на поток трафика. Это не должно вызывать удивления, поскольку среднее время обнаружения и восстановления меньше, чем срок действия правил потока. Например, в условии 5 потеря пакетов по-прежнему равна нулю. Правила потока, уже установленные C2 в S4, все еще действуют, но на основе шести условий они будут заменены новыми правилами потока по истечении срока действия C1 или C3.

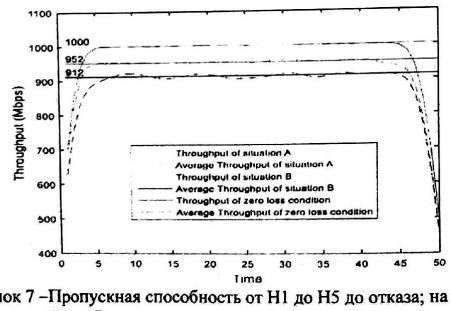
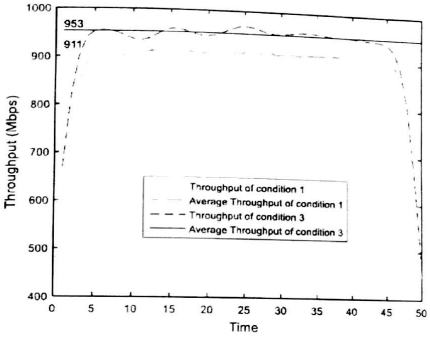


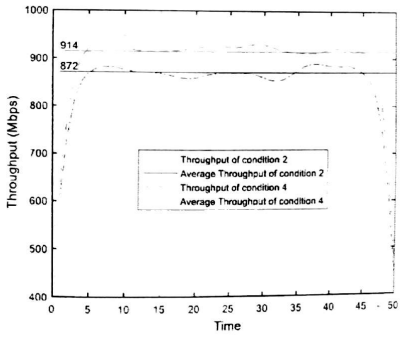
Рисунок 7 –Пропускная способность от H1 до H5 до отказа; на основе ситуаций A и B и ситуации потери нулевого соединения

Из рисунка 7 видно, что не все контроллеры смогли восстановиться после сбоя.

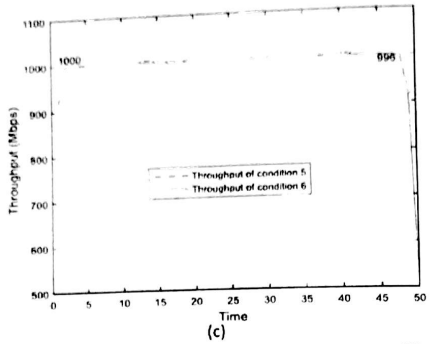
После восстановления после сбоя, необходимо оценить качество линии передачи. Оценка производительности производится путем сравнения пропускной способности в мегабитах в секунду в течение пятидесяти секунд. В нашем тестовом сценарии H1 отправляет трафик в H5 с помощью инструмента Iperf, поставляемого со средой Mininet. На графиках показана пропускная способность сети.



(a)



(b)



(c)

Рисунок 8 – Пропускная способность и средняя производительность от H1 до H5 после восстановления в условиях 1 и 3 в условиях 2 и 4 в условиях 5 и 6

Цель этих расчетов – предложить новый метод, называемый ReliableDistributed SDN (RDSDN), для управления надежностью в сети SDN. RDSDN рассматривает надежность плоскости данных и уровня управления вместе и объединяет координацию контроллеров и надежность плоскости данных для реализации действий по восстановлению после сбоев. Сначала для повышения точности действий RDSDN применяется параметр надежности, учитывающий некоторые метрики, такие как нагрузка, включая количество узлов плоскости данных, их степени и различные коэффициенты потерь в подключенных и взаимосвязанных каналах. Распределение Вейбулла для различных топологий SDN построено с учетом вышеупомянутых параметров для сравнения. Алгоритм поиска координатора в нашем методе RDSDN выбирает контроллер SDN поверх самой надежной инфраструктуры в качестве координатора. Его задача - контролировать другие контроллеры и брать на себя их роль в случае сбоев. Затем он реализует стратегии восстановления, управляя ролью ведущего / ведомого контроллеров, которые все еще активны. Доказательство концепции предложенной схемы было реализовано и протестировано. Удовлетворительные результаты в отношении времени восстановления после сбоя, потери пакетов, задержки и пропускной способности показывают, что с помощью метода RDSDN производительность и надежность всей сети были улучшены.

ЗАКЛЮЧЕНИЕ

В выпускной квалификационной работе были исследованы методы надежной и безопасной передачи данных в программно-конфигурируемых сетях. Исследуемый метод позволит с высокой надежностью передавать информацию в программно-конфигурируемых сетях.

В процессе работы были решены следующие задачи.

Проведен анализ источников литературы. После проведения анализа источников литературы выяснилось, что авторы источников лишь поверхностно прошли по методам надежности передачи данных в программно-конфигурируемых сетях. Задачей данной работы, выявить и доказать, что выбранный метод самый надежный из всех предложенных.

Основные преимущества метода RDSDNнад другими это:

- отказоустойчивая сеть;
- быстрота восстановления после сбоя/отказа сети;
- скорость передачи информации;
- качество передаваемой информации в программно-конфигурируемых сетях.

Проведены теоретические исследования в области применения технологии SDN.

Рассмотрено, что такое программно-конфигурируемые сети и как они работают. Показать уровни управления сетями SDNи как они функционируют между собой. Изучить проблему исследования современных проблем обеспечения безопасности и надежности в сетях SDN. Объяснить необходимость межсетевого экранирования в программно-конфигурируемых сетях, и каким образом межсетевые экраны связаны с коммутаторами.

Произведен анализ повышения надежности в программно-конфигурируемых сетях.

Дано объяснение необходимости роли контроллер в сетях SDN. Определены типы контроллеров:

- центральный;
- распределенный.

Рассмотреть полностью распределенные контроллеры:

- Onix;
- SmartLight;
-DISCO;
- ElastiCon;
- ICONA.

Сравнение этих контроллеров, а именно их принципом работы, надежностью передачи данных, схемой их расположения в сетях. Подытоживайте всех методов обеспечения надежности и сведение всех методов в таблицу 2.2. Сравнение всех предложенных методов с методов обеспечения надежности с методом RDSDN.

Построение сети при методе обеспечения надежности RDSDNподразумевает разделение сети на более мелкие подсети и назначение в этих сетях по одному контроллеру. Таким образом получается, что сеть, управляется одним контроллером, а более мелкие сети управляются своим контроллером SDN. Таким образом, сеть менее подвержена отказом узлов передачи или контроллеров.

Исследованы методы обеспечения надежности.

Произвели расчет показателя надежности, чтобы понять какой метод самый надежный. Выяснилось это уже в математическом расчете, то есть, наглядный пример. Критерии, которые учитывались при расчете показателя надежности:

- время задержки;
- время восстановления после сбоя;
- пропускная способность.

Был представлен перечень необходимого программного обеспечения, который использовался для эмуляции сети и использовался для получения численных данных.

После полученных результатов, данные были сведены в графики для более наглядного доказательства и сравнения всех контроллеров обеспечения надежности. По полученным данным видно, что метод RDSDNболее надежный чем все остальные методы.

Научная новизна заключается в определении более надежного метода передачи данных в программно-конфигурируемых сетях, которые подтверждаются математическими расчетами и моделированием в системе MathLab, где была сэмулирована сеть для подтверждения расчетных данных.

Теоретическая значимость заключается в выборе методов обеспечения надежности в программно-конфигурируемых сетях.

Практическая значимость заключается в определении наилучшего алгоритма обеспечения надежности при полученных расчетах и сравнении с ранее предложенными методами с учетом параметров времени восстановления, время задержки и скорость передачи.

СПИСОК РАБОТ, ОПУБЛИКОВАННЫХ АВТОРОМ ПО ТЕМЕ ДИССЕРТАЦИИ

1 Красулин Г.А. Методика расчета показателя надежности сетей SDN / Г.А. Красулин // Инженерные технологии: химия, биология, медицина и информационные технологии в промышленности.

2 Красулин Г.А. Программно-конфигурируемые сети / Г.А. Красулин // Международная научно-практическая конференция «Избранные вопросы науки XXI».

3 КрасулинГ.А. SDNNetworks / Г.А. Красулин // Студенческий вестник.

4 Красулин Г.А. Сети SDN /Г.А. Красулин // Информационные технологии и когнитивная электросвязь.