

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О.В. Непомнящий
«__» _____ 2023 г.

БАКАЛАВРСКАЯ РАБОТА

090301 Информатика и вычислительная техника

Изучение сетевого протокола под языком C++ в Omnet++

Руководитель	_____	_____	доцент, канд. техн. наук	Ф.А. Казаков
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Выпускник	_____	_____		И.И. Сошников
	<i>подпись</i>	<i>дата</i>		
Нормоконтролёр	_____	_____	доцент, канд. техн. наук	Ф.А. Казаков
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	

Красноярск 2023

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О.В. Непомнящий

« ___ » _____ 2022 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

Красноярск 2022

Студенту _____ Сошникову Игорю Ивановичу
фамилия, имя, отчество

Группа _____ КИ19-08Б _____ Направление (специальность) _____ 09.03.01
номер код

_____ Информатика и вычислительная техника
наименование

Тема выпускной квалификационной работы: _____ Изучение сетевого протокола
под языком С++ в Omnet++

Утверждена приказом по университету № _____ 4765/С от _____ 23.03.2023

Руководитель ВКР: _____ Казаков Ф.А., канд. техн. наук,
инициалы, фамилия, учёная степень, должность, место работы

_____ доц. НУЛ ТС кафедры ВТ, ИКИТ СФУ

Исходные данные для ВКР:

_____ Задание на ВКР.

Перечень разделов ВКР:

_____ 1) Анализ предметной области;

_____ 2) Протокол множества деревьев;

_____ 3) Разработка проекта.

Перечень графического материала: _____ Презентация в формате

_____ Microsoft PowerPoint.

Руководитель ВКР

подпись

_____ Ф.А. Казаков

инициалы, фамилия

Задание принял к исполнению

подпись

_____ И.И. Сошников

инициалы, фамилия

_____ «22» _____ 12 _____ 2022 г.

дата

РЕФЕРАТ

Выпускная квалификационная работа по теме «Изучение сетевого протокола под языком С++ в Omnet++» содержит 63 страниц текстового документа, 31 иллюстрации, 17 таблиц, 10 приложений, 14 использованных источников.

КОМПЬЮТЕРНАЯ СЕТЬ, КОММУТАЦИЯ, МАШРУТ, КАНАЛЬНЫЙ УРОВЕНЬ, ТОПОЛОГИЯ, STR, ПЕРЕДАЧА ДАННЫХ, ПРОТОКОЛ.

Целью данной работы стала разработка метода построения оптимальной топологии для передачи трафика по сети с использованием заблокированных каналов связи.

Для достижения поставленной цели решались следующие задачи:

- обзор существующих решений в области построения логических сетевых топологий;
- разработать метод оптимальных маршрутов для протокола множества деревьев;
- снизить загруженность корневого узла и корневых каналов связи;
- увеличить скорость потока передачи данных.

Была проанализирована область сетей передачи данных, выявлены направления, которые имеют свои недостатки – передача кадров на канальном уровне. Были поставлены задачи, решение которых позволит избавиться от выявленных недостатков. Протокол позволит передавать данные по сети с большей скоростью, а значит за меньшее время, распределяя нагрузку равномерно по всей топологии.

СОДЕРЖАНИЕ

Введение.....	4
1 Анализ предметной области и определение целей исследования	5
1.1 Модель OSI	5
1.1.1 Передача данных.....	7
1.1.2 Маршрутизация.....	10
1.1.3 Закольцованные сети.....	14
1.2 Иерархическая модель сети	15
1.3 Spanning tree protocol	16
1.3.1 Принцип работы.....	17
1.3.2 RSTP	21
1.3.3 Недостатки STP.....	23
1.4 Оптимизация передачи на канальном уровне.....	26
1.4.1 PVST	27
1.4.2 SSS	29
1.4.3 IFS.....	30
2 Протокол множества деревьев.....	33
2.1 Область применения	33
2.2 Основные понятия.....	34
2.3 Принцип построения деревьев.....	35
2.4 Состояния и роли портов.....	37
2.5 База данных коммутатора	38
2.6 Формат кадров.....	40
2.7 Алгоритмы работы коммутатора.....	42
2.8 Борьба с broadcast-штормами.....	42
2.9 Поддержка отношений	43
3 Разработка проекта.....	44
3.1 Полученные результаты	44

Заключение	51
Список использованных источников	52
Приложение А Включение коммутатора.....	54
Приложение Б Предложение об установке связи	55
Приложение В Действия при потере канала связи.....	56
Приложение Г Действия при подключении канала.....	57
Приложение Д Действия при необходимости отправки данных	58
Приложение Е Обработка кадра «Конфигурация дерева»	59
Приложение З Обработка кадра «Готовность».....	60
Приложение И Обработка кадра «Предложение Root»	61
Приложение К Обработка кадра «Потеряна Root-связь»	62
Приложение Л Обработка кадра «Данные».....	63

ВВЕДЕНИЕ

Для гарантирования надежности сети часто используют избыточные каналы связи, однако они не должны создавать петли из-за особенностей передачи трафика. Для предотвращения петель в сетевой топологии коммутаторов можно отключить некоторые каналы связи, чтобы избежать возможных кольцевых соединений и создать логическую топологию дерева на основе физической топологии. В результате возрастает нагрузка на каналы связи, расположенные ближе к корневому коммутатору, и возникает множество заблокированных физических каналов связи, которые могут быть использованы в дальнейшем.

Актуальность исследования заключается в том, что на сегодняшний день нет какого-либо универсального способа решения этих проблем, которые с ростом масштабов вычислительных сетей становятся все более заметными. Поэтому создание метода, позволяющего распределить нагрузку по сети, тем самым увеличив её общую пропускную способность, является достаточно важной и практически значимой задачей.

В связи с этим, целью данной работы стало разработка метода построения оптимальной логической топологии для передачи трафика по сети с использованием заблокированных каналов связи.

Для достижения поставленной цели решались следующие задачи:

- обзор существующих решений в области построения логических сетевых топологий;
- разработать метод оптимальных маршрутов для протокола множества деревьев;
- снизить загруженность корневого узла и корневых каналов связи;
- увеличить скорость потока передачи данных.

1 Анализ предметной области и определение целей исследования

1.1 Модель OSI

Сетевая эталонная модель OSI (Open Systems Interconnection Basic Reference Model) представляет собой семиуровневый стек, каждый уровень из которого выполняет свои функции и обеспечивает взаимодействие с соседними уровнями. Модель OSI стала большим шагом при проектировании концепции современных сетей. Она привнесла идею общей модели протоколов, расположенных на различных уровнях и определяющих взаимодействие между сетевым оборудованием и программным обеспечением. Однако, в связи со своей сложностью, модель OSI осталась нереализованной на практике и является чисто теоретической, но она помогает разрабатывать протоколы и приложения.

Стек протоколов, который применяется практически во всех современных сетях – это протокол TCP/IP (Transport Control Protocol/Internet Protocol). Соответствие уровней протоколов приведено в таблице 1.

Таблица 1 – Сравнение стеков протоколов

Модель OSI	Модель TCP/IP
7. Уровень приложений	Уровень приложений
6. Уровень представления	
5. Уровень сессии	
4. Транспортный уровень	Транспортный уровень
3. Сетевой уровень	Сетевой уровень
2. Канальный уровень	Канальный уровень
1. Физический уровень	

Функции, выполняемые на уровнях моделей OSI:

7. Уровень приложений – обеспечивает доступ прикладных процессов к сетевым услугам, а также обеспечивает взаимодействие между приложениями.

Протоколы, работающие на этом уровне: HTTP (Hyper Text Transfer Protocol), SMTP (Simple Mail Transfer Protocol) , POP3 (Post Office Protocol Version 3) , TELNET (TErminAL NETwork).

6. Уровень представления – отвечает за кодирование/декодирование данных. Запросы с уровня приложений на этом уровне преобразуются в формат передачи по сети, а запросы из сети преобразуются в формат, понятный приложениям. Один из протоколов, работающих на уровне представления – SSL (Secure Socket Layer).

5. Уровень сессии – уровень отвечает за установку и поддержание сеанса связи, его синхронизацию, а также поддержание упорядоченного взаимодействия при обмене данными между объектами уровня представления. Протоколы уровня сессий: PPTP (Point-to-Point Tunneling Protocol), L2TP (Layer 2 Tunneling Protocol), SMPP (Short message peer-to-peer protocol).

4. Транспортный уровень – обеспечивает доставку данных. Предназначен для межконцевого взаимодействия (точка-точка). Транспортные функции зависят от сетевого сервиса и могут включать отображение транспортного адреса в сетевой адрес, управление потоками на отдельных соединениях, обнаружение и исправление ошибок качества сервиса, сегментирование, блокирование и сцепление данных. PDU (Protocol Data Unit) – фрагмент данных, с которыми работает транспортный уровень – TCP-сегмент или UDP-датаграмма. Основные протоколы транспортного уровня являются TCP (Transmission Control Protocol) и UDP (User Datagram Protocol).

3. Сетевой уровень – предназначен для адресации сообщений и перевод логических адресов в физические. На этом уровне определяется маршрут от объекта-отправителя к объекту-получателю и обеспечивается независимость от особенностей маршрутизации и ретрансляции, связанных с установкой и использованием сетевого соединения. PDU сетевого уровня – пакет. Основной протокол сетевого уровня – IP (Internet Protocol версий 4 и 6),

2. Канальный уровень – обеспечивает функциональные и процедурные средства для установления, поддержания и расторжения канальных соединений между сетевыми объектами и передачи блоков данных. Так же выполняет функции обнаружения и исправления ошибок физического уровня, управляет соединением физических каналов передачи данных. PDU канального уровня – кадр. На сегодняшний день самым распространенным протоколом и технологией канального уровня стал Ethernet.

1. Физический уровень – обеспечивает передачу неструктурированного потока бит по физическим каналами связи, отвечает за кодирование данных и синхронизацию битов.

Три нижних уровня являются сетезависимыми, то есть протоколы этих уровней тесно связаны с технической реализацией сети и используемым сетевым оборудованием. Адресация и маршрутизация сетевого трафика происходит на сетевом и канальном уровнях, и зависит от используемого оборудования. На сетевом уровне работают маршрутизаторы и коммутаторы третьего уровня, на канальном – обычные коммутаторы второго уровня. Для обеспечения высокой производительности сети и её устойчивости используются протоколы маршрутизации, которые позволяют настроить логическую адресацию в топологии сети на третьем (сетевом) уровне, осуществить балансировку нагрузки на сетевом уровне.

Протоколы канального уровня предназначены для того, чтобы обезопасить этот уровень от появления петель.

1.1.1 Передача данных

Передача данных можно происходить тремя способами:

- unicast – один источник, один приемник;
- broadcast – один источник, приемниками являются все остальные устройства в подсети (сегменте);

– multicast – один источник, несколько приемников, принадлежащих к одной группе.

Чтобы передать данные между двумя узлами в сети, их необходимо обработать на каждом из уровней модели TCP/IP. Сначала, на уровне приложения определяется объем данных, которые необходимо передать. На транспортном уровне они разбиваются на сегменты, на сетевом уровне – формируются пакеты, затем пакеты упаковываются в кадры на канальном уровне и на физическом уровне биты кадров кодируются в последовательности оптических или электрических сигналов.

Процесс обработки данных на каждом уровне называется инкапсуляцией. При передаче сообщения на каждом уровне модели TCP/IP к фрагменту данных добавляются необходимые заголовки, в зависимости от конкретного протокола, и передаются на нижележащий уровень.

На физическом уровне данные передаются по физическим каналам связи в виде последовательностей физических или электрических сигналов и детектируются на физическом уровне стека протоколов узла-приемника. После этого данные передаются от нижних уровней к верхним, на каждом уровне стека соответствующие протоколы считывают заголовки своих уровней. Этот процесс называется декапсуляцией.

Обработка данных при передаче по стеку протоколов TCP/IP изображена на рисунке 1.

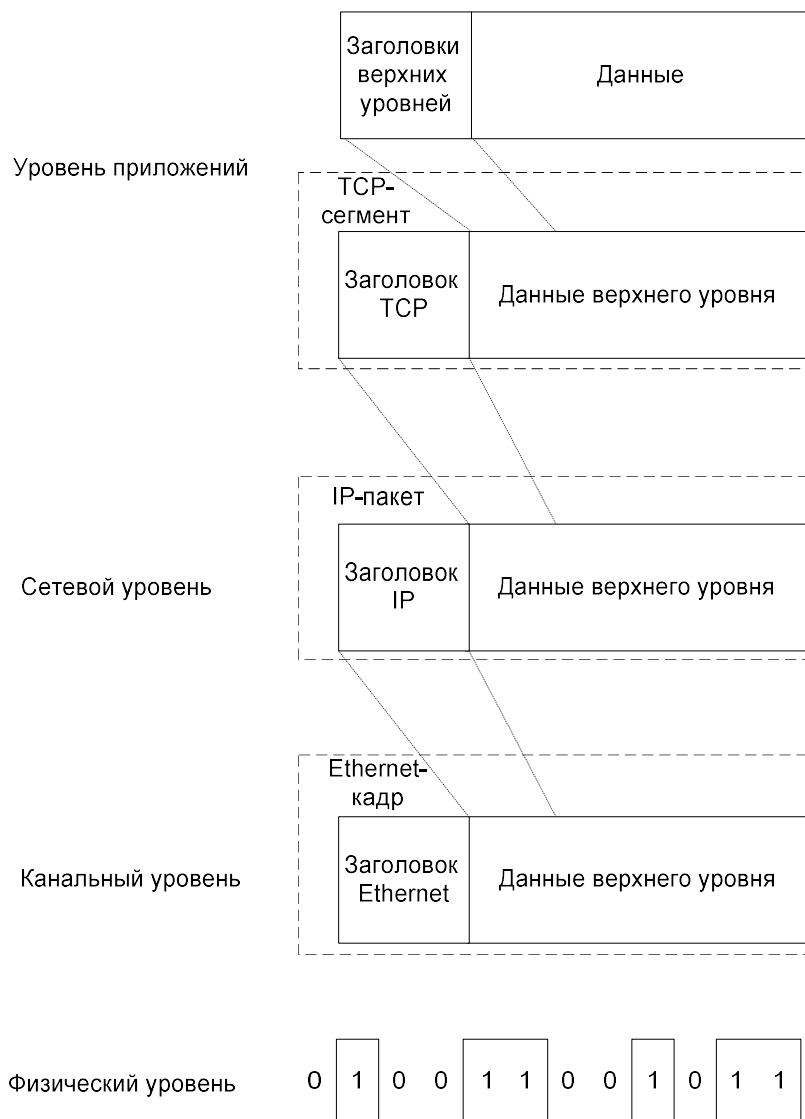


Рисунок 1 – Инкапсуляция данных

Для того, чтобы попасть от узла-отправителя к узлу-источнику, кадр проходит через некоторое количество коммутаторов или маршрутизаторов. Каждому из этих сетевых устройств необходимо обработать принятое сообщение, глубина обработки сообщения зависит от уровня, на котором работает данное устройство. Маршрутизаторы работают на третьем (сетевом) уровне, и для обработки принятого кадра им требуется обработать заголовки канального уровня (сравнить MAC-адрес узла-назначения), передать декапсулированный пакет на уровень выше и обработать заголовки сетевого уровня. После этого либо отбросить пакет, либо пересчитать некоторые

заголовки, контрольные суммы и инкапсулировать пакет в кадр для последующей отправки. Коммутаторы обычно работают на втором (канальном) уровне. При обработке кадра они только смотрят заголовки этого уровня, работают с ними и отправляют кадр дальше к узлу назначения.

1.1.2 Маршрутизация

Процесс выбора маршрута через множество сетевых устройств от узла-источника к узлу назначения называется маршрутизацией. Она может проходить на канальном и сетевом уровне. Так как кадры обрабатываются на каждом сетевом устройстве на маршруте, а пакет только на устройствах третьего уровня, маршрутизация на канальном и сетевом уровне являются разными процессами.

На сетевом уровне определяется последовательность маршрутизаторов, через которые следует отправить пакет, чтобы достичь сети, к которой принадлежит узел назначения. Чем меньше сетевых устройств стоит на пути пакета, тем меньше будет время задержек на его обработку на каждом устройстве, чем выше пропускная способность каналов связи, тем быстрее пакет будет достигать цели. Отсюда можно определить задачу маршрутизации на сетевом уровне – поиск наиболее оптимального пути через маршрутизаторы в компьютерной сети.

Маршрут на канальном уровне в одном сегменте сети существует только один, так как для избегания циклов между сетевым оборудованием из физической топологии сети строится логическая топология-дерево. О необходимости этого будет написано далее.

Рассмотрим процесс адресации данных при передаче их по сети. Возьмем топологию, представленную на рисунке 2.

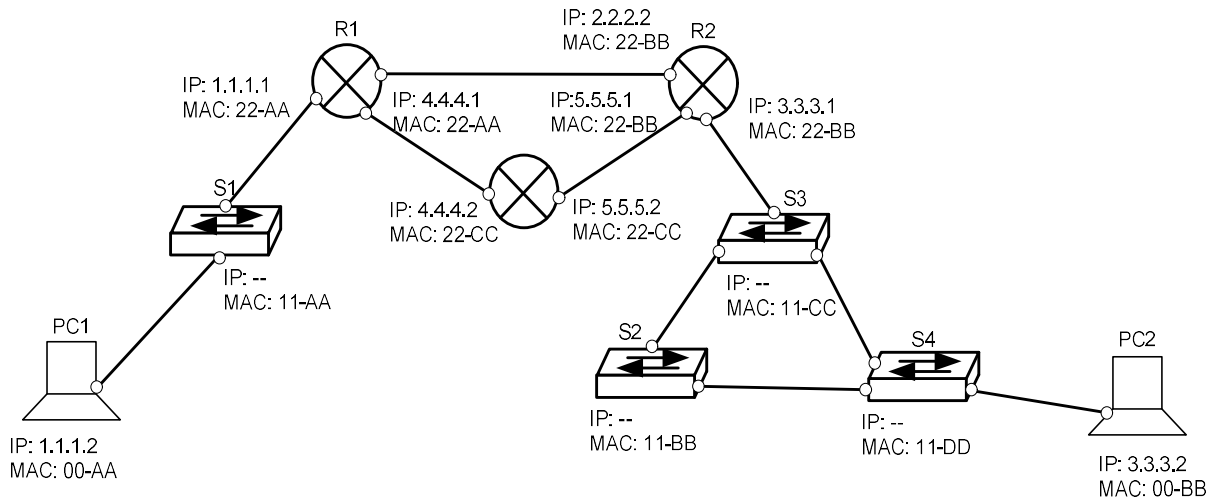


Рисунок 2 – Топология сети

За исходные данные примем, что таблицы коммутации на коммутаторах заполнены правильно, протоколы маршрутизации на сетевом и канальном уровне отработали корректно и маршрутизация между сетями настроена. В данном примере нас интересует адресация на сетевом и канальном уровнях. Проведем их анализ отдельно и обратим внимания только на необходимые в этом случае поля заголовков – IP-адреса и MAC-адреса, а также поле TTL (время жизни пакета) которое будет изменяться по ходу движения пакета от отправителя к адресату (Рисунок 3).

На сетевом уровне маршрут пакета выглядит следующим образом:

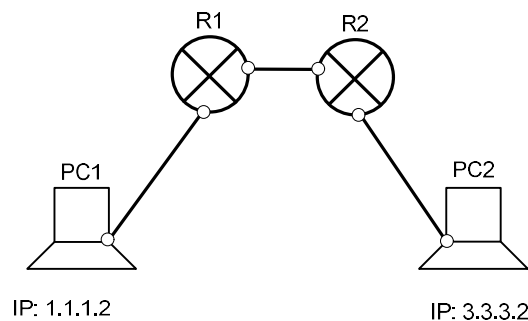


Рисунок 3 – Маршрут на сетевом уровне

Пакет обрабатывается только устройствами, работающими на сетевом уровне, в данном случае – маршрутизаторами. На рисунке 4 показаны структура и значения заголовков пакета на каждом из этапов его передачи до узла назначения.

IP-Адрес отправителя	IP-Адрес получателя	TTL	Поле данных
1.1.1.2	3.3.3.2	128	DATA

IP-Адрес отправителя	IP-Адрес получателя	TTL	Поле данных
1.1.1.2	3.3.3.2	127	DATA

IP-Адрес отправителя	IP-Адрес получателя	TTL	Поле данных
1.1.1.2	3.3.3.2	126	DATA

Рисунок 4 – Структура пакета

Из рисунка 4 видно, что при пересылке пакета, у него изменяется только поле TTL (Time-to-live). Значение этого поля уменьшается на единицу, когда пакет проходит через маршрутизатор, и когда оно достигает нуля, этот пакет отбрасывается. Такой механизм используется для того, чтобы пакеты не пересылались бесконечно, если в сети вдруг возникнет кольцевой путь из маршрутизаторов. В случае адресации типа broadcast или multicast, в поле IP-адрес получателя указывается либо широковещательный адрес сети, либо групповой адрес соответственно.

На канальном уровне идет оперирование кадрами, в заголовке которого используется MAC-адреса устройств, между которыми происходит передача кадра. Причем, в процессе его обработки на коммутаторах с кадром не происходит никаких изменений, он лишь направляется согласно таблице

коммутации. На маршрутизаторе MAC-адреса источника и приемника в кадре изменяются и пересчитывается контрольная сумма кадра (Рисунок 5).

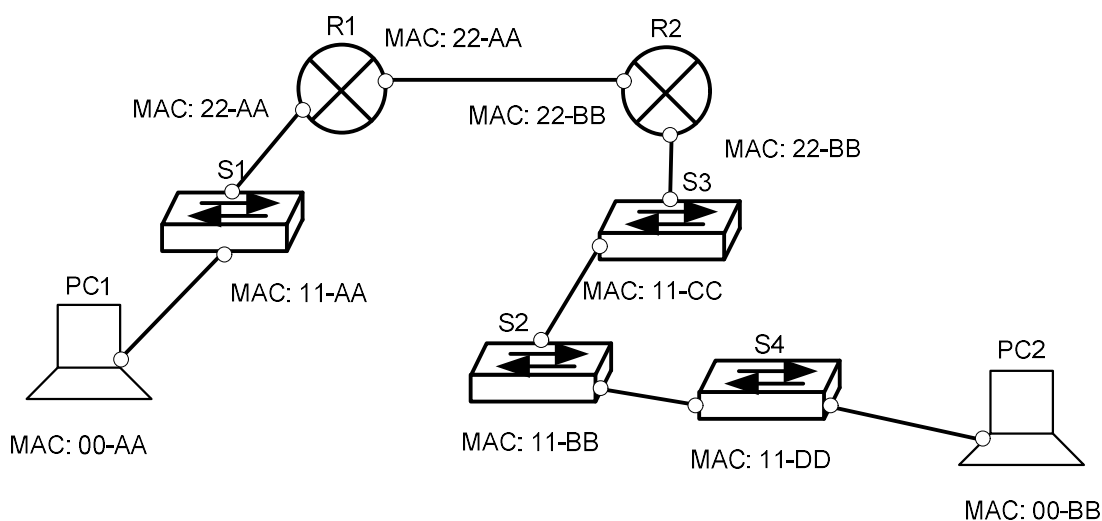


Рисунок 5 – Маршрут на канальном уровне

Во время прохождения маршрута от PC1 к PC2 кадр изменяется два раза во время обработки на маршрутизаторах (Рисунок 6).

MAC-адрес отправителя	MAC-Адрес получателя	Поле данных
00-AA	22-AA	Ethernet-пакет
MAC-адрес отправителя	MAC-Адрес получателя	Поле данных
22-AA	22-BB	Ethernet-пакет
MAC-адрес отправителя	MAC-Адрес получателя	Поле данных
22-BB	00-BB	Ethernet-пакет

Рисунок 6 – Структура кадра на маршруте

В отличие от пакета, кадры не имеют поля TTL, таким образом, появление колец на втором уровне вызывает лавинное заполнение каналов связи, что парализует работу сегмента сети. Если у коммутаторов в таблице коммутации нет выходного интерфейса для MAC-адреса узла-назначения, он разошлет кадр по всем своим интерфейсам, кроме того, на который кадр пришел. В случае нескольких 24- или 48-портовых коммутаторов, соединенных каскадом, количество кадров возрастает экспоненциально.

1.1.3 Закольцованные сети

Появление колец в сети опасно тем, что в пределах них возможно лавинное образование и передача пакетов или кадров. Это явление называется broadcast-штормом, и если в случае колец из маршрутизаторов, одноадресный пакет будет отброшен, то при групповой или широковещательной рассылке пакет может бесконечно дублироваться и нагружать собой каналы связи и сетевое оборудование.

Циклы на канальном уровне становятся еще опаснее. Так как у кадров нет понятия «время жизни», при экспоненциальном росте их количества они не отбрасываются и используют всю пропускную способность сети, парализуя её работу. Единственный способ восстановить работу сети в этом случае – физическое выключение активного сетевого оборудования, так как возможность удаленного управления оборудованием становится невозможной.

Для борьбы с кольцами на втором и третьем уровнях модели OSI используются протоколы маршрутизации и прямая настройка. На сетевом уровне это такие протоколы как OSPF (Open shortest path first), RIP (Routing internet protocol), EIGRP (Enhanced Interior Gateway Routing Protocol). На канальном уровне используется протокол STP (Spanning tree protocol) или его расширения, в основе которых лежит алгоритм связующего дерева. О работе этого протокола подробнее рассказывается далее.

1.2 Иерархическая модель сети

Модель современных вычислительных локальных сетей состоит из трех уровней, каждый из которых выполняет определенные функции, который помогают наиболее полноценно использовать потенциал и возможности сетей. Модель состоит из трех уровней (Рисунок 7):

1. Базовый уровень, ядро. На уровне ядра необходима скоростная и отказоустойчивая пересылка большого объема трафика без появления задержек, поэтому на этом уровне используют наиболее производительное и дорогостоящее оборудование с обязательным резервированием каналов.

2. Уровень распределения. На уровне распределения также необходима высокая пропускная способность и возможность агрегации трафика уровня сети. На уровне происходит маршрутизация трафика на третьем уровне модели OSI (уровне сети), определение широковещательных и многоадресных доменов, реализация сетевых политик.

3. Уровень доступа. На уровне доступа происходит подключение к сети конечных пользователей. Агрегация пользовательского трафика для передачи его на уровень распределения, реализация безопасности и политик доступа.

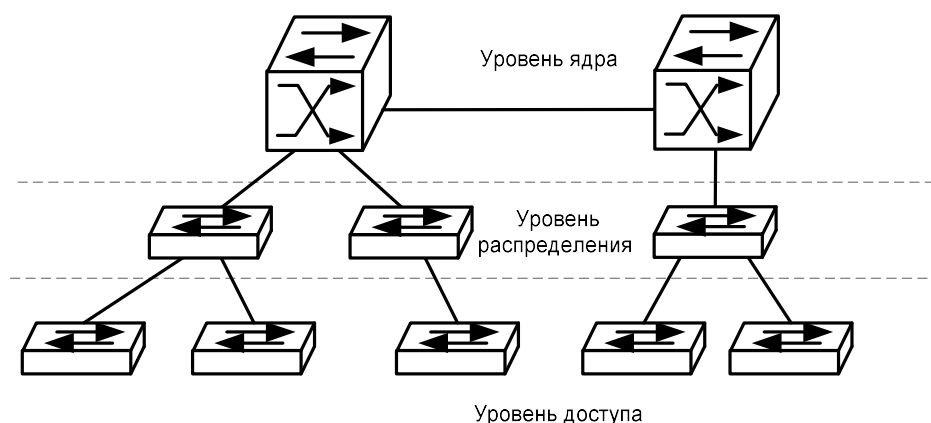


Рисунок 7 – Иерархическая модель сети

Для маршрутизации трафика на сетевом уровне необходимо на каждом маршрутизаторе распаковывать принятые кадры, обрабатывать пакеты на третьем уровне, определять узел назначения для него, упаковывать пакет в кадр и отправлять его дальше по каналам передачи данных. Все эти манипуляции с кадром требуют процессорного времени и создают задержку при передаче данных, чего необходимо избежать на уровне ядра. Чтобы избежать этого, уровень ядра трафик необходимо маршрутизировать на втором уровне (канальный уровень модели OSI). В таком случае кадры, попадающие интерфейсы коммутаторов на уровне ядра будут сразу отправлены до следующего узла, без обработки их содержимого. Вся необходимая для маршрутизации информация находится в заголовках кадра.

На данный момент существуют несколько вариантов построения логических топологий на канальном уровне модели OSI.

1.3 Spanning tree protocol

Основная задача протокола – предотвратить появление петель на втором уровне в топологии сети. Чтобы добиться этого, необходимо логически заблокировать некоторые физические каналы, необходимые для резервирования, но в которых нет необходимости, когда сеть работает стабильно и не происходит изменений в её топологии. Протокол STP постоянно проверяет сеть на предмет появления новых каналов связи или сбоя уже существующих, в случае изменения топологии сети, коммутаторы автоматически реконфигурируют свои порты для того, чтобы предотвратить потерю каналов связи и возникновения петель.

1.3.1 Принцип работы

Протокол использует алгоритм STA (Spanning Tree Algorithm), результатом работы которого становится граф в виде дерева, в котором коммутаторы выстроены в иерархической структуре. Для обмена информацией коммутаторы используют кадры BPDU (Bridge Protocol Data Unit) двух типов: конфигурационные кадры и кадры о реконфигурации.

1. Конфигурационные кадры – регулярно рассылаются корневым коммутатором и ретранслируются другими и используются для построения топологии.

2. Кадры уведомления о реконфигурации – рассылаются любыми коммутаторами в случае изменения топологии сети.

Поля конфигурационного кадра описаны в таблице 2.

Таблица 2 – Формат кадра конфигурации

Название	Размер
Идентификатор версии протокола STA	2 байта
Версия протокола STP	1 байт
Тип BPDU	1 байт
Флаги	1 байт
Идентификатор корневого коммутатора	8 байт
Расстояние до корневого коммутатора	4 байта
Идентификатор коммутатора	8 байт
Идентификатор порта	2 байта
Время жизни сообщения	2 байта
Максимальное время жизни сообщения	2 байта
Интервал, через который посылаются пакеты BPDU	2 байта
Задержка смены состояния	2 байта

Топология без петель строится следующим образом. Сначала выбирается корневой коммутатор, который является корневым узлом дерева и считается центром сети. Выбор основывается на идентификаторе коммутатора. Идентификатор получается способом сложения приоритета и MAC-адреса устройства (Рисунок 8).

MAC-адрес: Приоритет:
FF-EE-DD-CC-BB-AA 32768

Идентификатор
коммутатора:
32768FFEEDDCCBBAA

Рисунок 8 – Построение идентификатора

Вначале, каждый коммутатор считает себя корневым и рассылает другим коммутаторам BPDU-уведомление, где объявляет об этом. Как только он получает BPDU-кадр, в котором идентификатор узла-отправителя меньше, чем его собственный, коммутатор перестает считать себя корневым и начинает рассылать BPDU-уведомления, в которых указывает идентификатор корня. В итоге во всех BPDU-уведомлениях начинает фигурировать только один идентификатор, обладатель которого и станет корневым коммутатором.

После того, как корневой коммутатор был найден, всем остальным необходимо выбрать только один порт, который будет направлен в сторону корня. Такой порт называется корневым портом. Чтобы понять, какой порт надо использовать, необходимо вычислить «стоимость» маршрута до корня через все порты (Таблица 3). Процесс определения стоимости выглядит следующим образом:

1. Корневой коммутатор рассылает BPDU со стоимостью маршрута 0 (поле Root Path Cost).
2. Коммутатор, который получает BPDU, смотрит на скорость своего порта, с которого пришел кадр и добавляет в поле Root Path Cost, стоимость, согласно таблице 2.
3. Если два порта имеют одинаковую стоимость, выбирается младший из них.

4. После этого BPDU рассылается нижестоящим коммутаторам с новой стоимостью и так далее.

Таблица 3 – Стоимость портов в протоколе STP

Скорость порта	Стоимость STP (802.1d)
10 Мбит	100
100 Мбит	19
1 Гбит	4
10 Гбит	2

Из каждого сегмента сети должен существовать только один выход по направлению к корневому коммутатору. Обычно это физический канал, который проходит между корневым и назначенным портами. После того, как на коммутаторе определены оба типа этих портов, остальные порты переводятся в состоянии «заблокированы», для предотвращения образования петель. Заблокированные порты ведут к корню, принимают BPDU, но не пересылают и не принимают трафик.

Итак, во время построения STP-дерева порты могут находиться в следующих состояниях:

- прослушивание – принимает входящие BPDU и начинает сам их отправлять;
- обучение – принимает все кадры (данные и BPDU), записывает MAC-адреса таблицу коммутации;
- пересылка – принимает и передает данные и BPDU;
- блокировка – канал блокирован, принимает только BPDU пакеты.

Для предотвращения слишком частого перестроения дерева топологии, когда возникают помехи на линиях связи и происходят потери сообщений BPDU в протоколе введено понятие таймеров (Таблица 4).

Таблица 4 – Таймеры протокола STP

Таймер	Назначение
Hello	Время между отправкой конфигурационных сообщений протокола STP. По умолчанию равно 2 секундам.
Forward delay	Длительность состояния прослушивания и изучения топологии.
Max age	Время хранения

При изменении в топологии сети, коммутатор, у которого появилась новая связь или была утрачена существующая, ожидает в течение периода времени max age, чтобы убедиться, что топология действительно была изменена. После этого он отправляет кадр уведомления к корневому коммутатору. Каждый коммутатор на маршруте должен уведомить пересылающий коммутатор о том, что кадр был получен. После того, как корневой коммутатор получил этот кадр, он рассылает по всей сети сообщение об изменении топологии. Получение этого сообщения информирует не корневые коммутаторы о том, что необходимо установить время обновления таблиц коммутации. После прохождения этого периода, таблица будет очищена и передача кадров данных восстановится. Пример распространения сообщений о реконфигурации показан на рисунке 9.

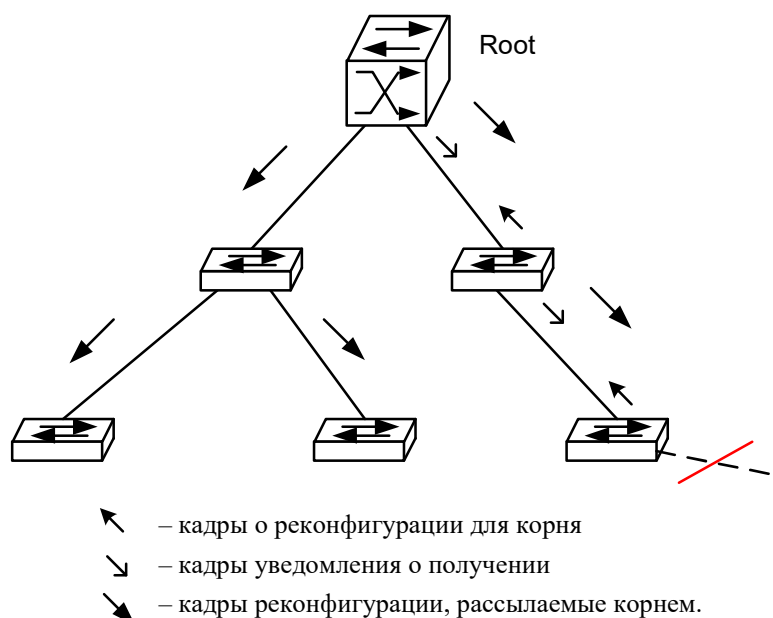


Рисунок 9 – Распространение информации о реконфигурации в STP

1.3.2 RSTP

Rapid Spanning Tree protocol – быстрый протокол STP. Является улучшенной версией протокола STP с уменьшенными временами сходимости и восстановления. Для уменьшения времени сходимости исходная спецификация стандарта 802.1D (стандартный протокол STP) была дополнена функциями Cisco Uplink Fast, Backbone Fast и Port Fast. Недостатками такого решения является то, что эти механизмы являются запатентованными и им требуется дополнительная настройка.

В протоколе RSTP используются 3 состояния портов, в отличие от оригинального протокола STP, состояния «блокировка» и «прослушивания» объединены в состояние «Сбрасывание».

Основные состояния портов:

1. Сбрасывание. Состояние порта, которое присуще как активной топологии, так и топологии в которой происходят синхронизация или изменения. Состояние сбрасывания предотвращает передачу кадров данных, тем самым разрываются физические петли в активной топологии.

2. Обучение. Это состояние порта присуще как активной топологии, так и топологии в которой происходят синхронизация или изменения. Состояние обучения позволяет кадрам данных заполнять таблицу MAC-адресов для ограничения передачи неизвестных одноадресных кадров.

3. Пересылка. Состояние порта присуще только стабильной активной топологии. В состоянии «пересылка» портами коммутаторов производится передача кадров данных. Во время синхронизации топологии процесс передачи кадров данных происходит только после процесса согласования.

В протоколе RSTP появилась возможность обратной связи, для объявления того, что сеть выполнила схождение.

Как говорилось ранее, в протокол STP был добавлен механизм Port Fast. Он позволяет портам коммутатора, которые подключены к рабочим станциям,

сразу после включения переходить в состояние пересылки, минуя промежуточные состояния. Однако, если по ошибке этот механизм будет применен к порту, с другой стороны которого находится коммутатор, это может привести к появлению петель в топологии.

Механизм работы Uplink Fast состоит в том, что заблокированные порты получают статус резервных. В случае обрыва основного канала связи, они мгновенно включаются, минуя стадии прослушивания и изучения топологии. Данная технология применяется на коммутаторах уровня доступа.

Технология Backbone Fast позволяет сократить время сходимости дерева почти в два раза при стандартном значении параметров задержки. Этот механизм применяется на всех трех уровнях. При разрыве связи на корневом порту, согласно протоколу STP, устройство начинает рассылать свои сообщения BPDU, для попытки объявить себя корневым коммутатором. Соседние устройства, получив на свои назначенные порты такой кадр, мгновенно опровергают его, отправляя ответный BPDU с правильным идентификатором корня.

Кроме этого, в протоколе RSTP изменена схема распространения уведомления о реконфигурации, в нем потеря связи не расценивается как изменение топологии, передача кадра происходит одноэтапно, без подтверждения и фактически инициатор сам заполняет сеть сообщениями о добавлении или удалении канала связи (Рисунок 10).

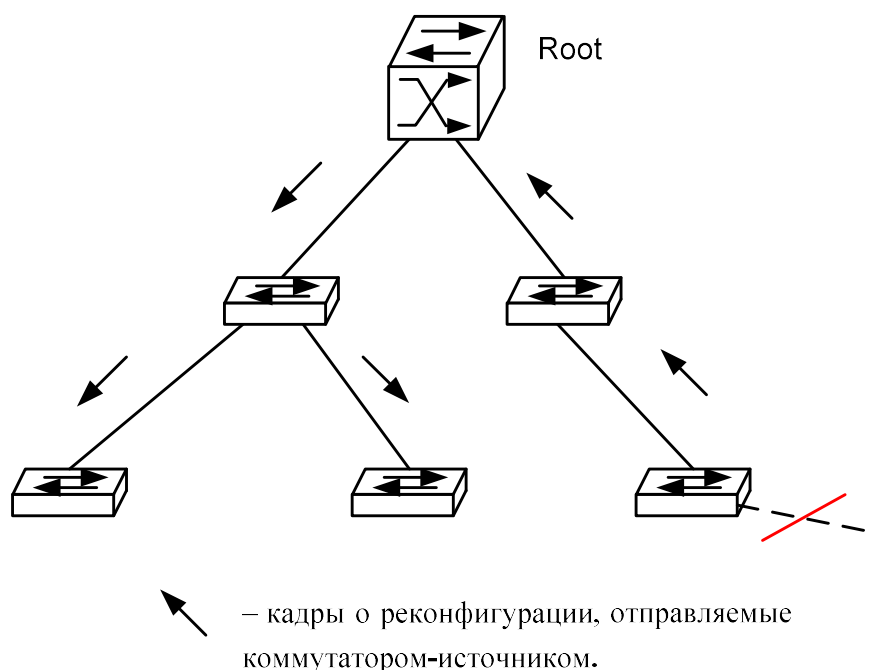


Рисунок 10 – Распространение информации о реконфигурации в RSTP

В протоколе RSTP не используется механизм подтверждения получения такого кадра, а также отсутствуют кадры реконфигурации, распространяемые корнем. Их роль исполняют кадры, отправляемые коммутатором-источником, после принятия которых, коммутаторы очищают записи в таблице коммутации, связанные с портами, куда был принят такой кадр, а также перенаправляют кадр на другие свои назначенные и корневые порты.

1.3.3 Недостатки STP

Протокол STP обладает большой инертностью. Это можно увидеть, как при начальном построении топологии, так и при реконфигурации существующего дерева.

При включении домена устройств, как быстро бы они ни сформировали дерево, порты не смогут передавать пользовательские данные в течение периода времени от 30 до 50 секунд при стандартных значениях таймеров. Это связано с переходом включенных портов из состояния «отключен» в состояние

«пересылка». То же самое происходит при разрыве топологии сети, с даже если сеть после изменения сошлась достаточно быстро, происходит ожидание по нескольким таймерам, прежде чем начнется передаваться пользовательский трафик.

О недостатках процесса реконфигурации при изменении состояний каналов дерева STP было рассказано в пункте 1.3.1

Помимо большого времени сходимости, одним из недостатков протокола STP является то, что весь трафик из разных пользовательских сетей проходит через корневой коммутатор, соответственно, чем ближе находится канал к корню, тем больше он нагружен.

Например, мы имеем следующую топологию из одиннадцати коммутаторов (рисунок 11).

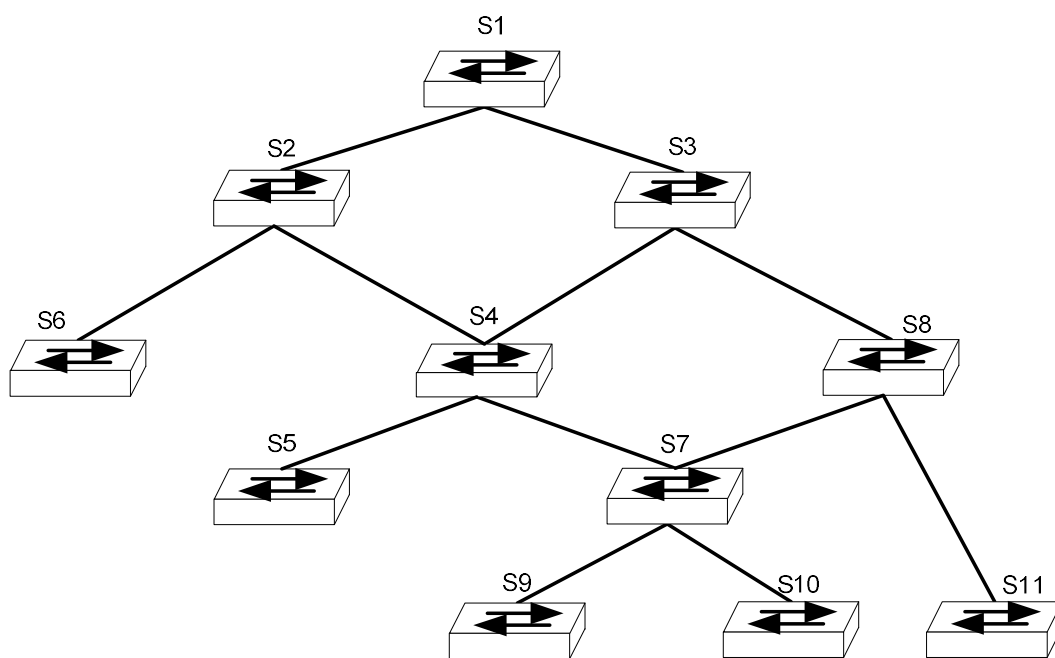


Рисунок 11 – Топология сети

Предположим, что при построении дерева в протоколе STP корневым будет выбран коммутатор S1, а все каналы имеют одинаковую скорость. После

рассылки BPDU кадров и перехода всех портов в конечное состояние дерево будет выглядеть следующим образом (Рисунок 12).

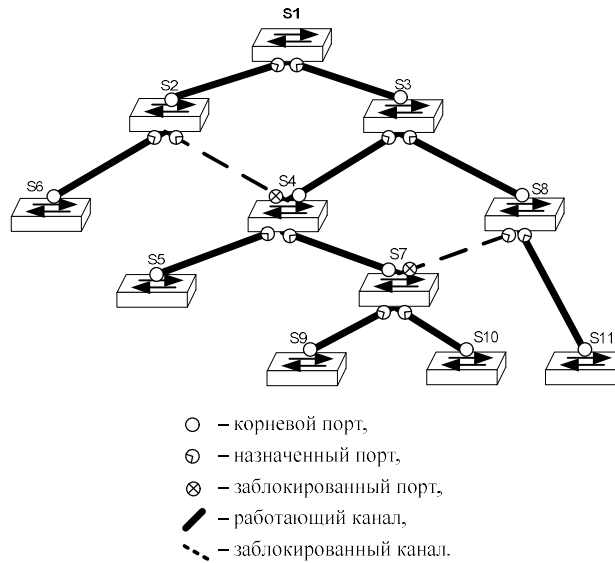


Рисунок 12 – Активная топология сети

Каналы между S2 и S4, а также между S7 и S8 заблокированы для предотвращения петель. Данный метод является эффективным для борьбы с петлями, однако он негативно сказывается на скорости передачи информации (Рисунок 13). Например, при необходимости передачи данных с компьютера PC1 к компьютеру PC2 трафик будет направлен по маршруту S6-S2-S1-S3-S4-S7-S9, хотя более коротким путем могла бы стать цепочка S6-S2-S4-S7-S9. В таком случае мы избавимся от трех лишних каналов связи, тем самым уменьшив их загруженность. Или же нам необходимо передать данные с компьютера PC2 к компьютеру PC3. Будет использован маршрут S9-S7-S4-S3-S8-S1, хотя лучшим в этом случае будет маршрут S9-S7-S8-S1 и мы сможем избавиться от трех каналов связи, что в два раза сократит маршрут передачи.

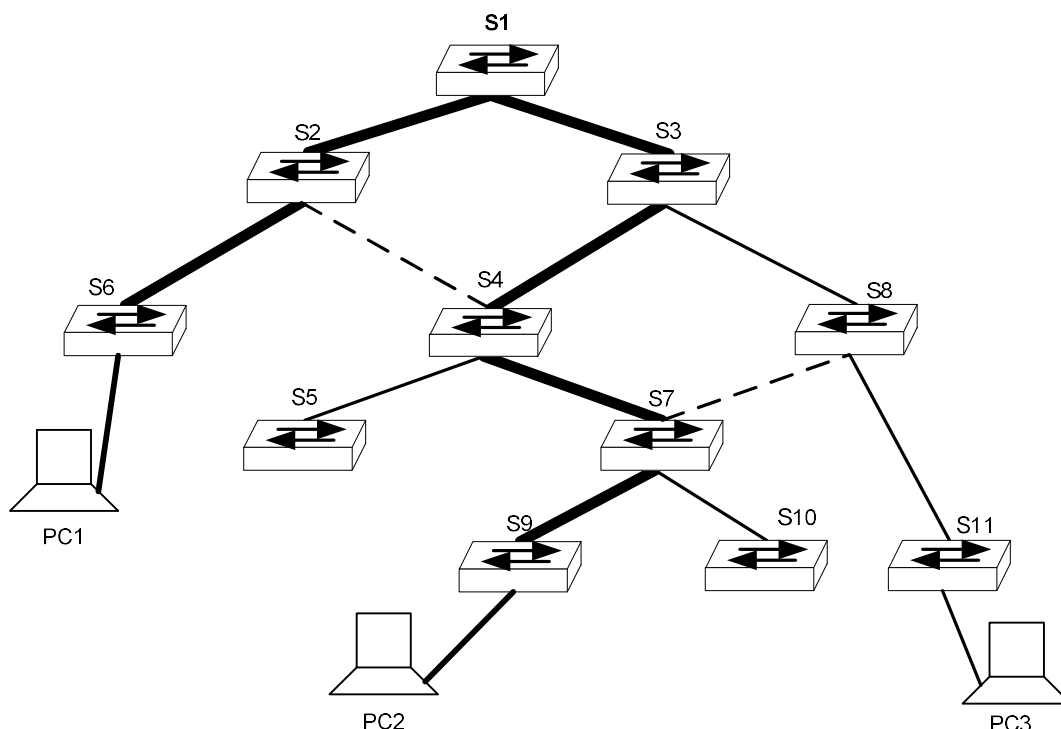


Рисунок 1.13 – Передача кадра от PC1 к PC2

Однако в данной ситуации - это невозможно, потому что протокол STP не позволяет использовать каналы связи S2-S4 и S7-S8, а вместо этого направляет весь трафик к корневому коммутатору S1, что загружает каналы S1-S2 и S1-S3. При большом количестве передаваемых данных в сети это может значительно сказаться на пропускной способности такой сети.

1.4 Оптимизация передачи на канальном уровне

Необходимо такое решение, которое бы позволяло использовать заблокированные связи в том случае, если это является оправданным с точки зрения передачи данных и разгрузки корневых каналов, а также было бы устойчиво с точки зрения отсутствия петель в сети. Есть несколько подобных вариантов расширения стандартного протокола STP.

Рассмотрим решение, разработанное компанией Cisco и применяющееся на практике, а также разработку иранских ученых – расширение протокола STP на базе метода shortcut switching strategy (метод сокращенных коммутаций).

1.4.1 PVST

Протокол Per-VLAN Spanning Tree Protocol (STP на каждый VLAN) является фирменным протоколом компании Cisco и не поддерживается другими производителями. В этом протоколе возможно построение виртуальных локальных сетей на физической топологии коммутаторов. Различные виртуальные сети могут быть использованы из соображений безопасности, для выделения отделов предприятия или разделения сетей по виду трафика – телефония и видеоконференции в одной сети, пользовательские данные в другой.

Так как для каждой виртуальной сети строится своя логическая топология, в каждой из которых собственное покрывающее дерево. Пользуясь этим, можно спроектировать виртуальные сети так, чтобы задействовать максимальное количество свободных каналов связи, разделить нагрузку между ними и оптимизировать передачу данных в сети. В каждой VLAN неизбежна блокировка каких-либо каналов связи, однако для каждой из них эти каналы будут своими, что позволит задействовать максимальное количество каналов связи и пропускной способности сети.

На рисунке 14а показан пример топологии сети. С работающим протоколом STP и настроенным корневым коммутатором S1, активная логическая топология представляет собой рисунок 14б. Каналы связи между S2 и S5, S4 и S3 заблокированы для избегания петель в сети, однако они могли бы быть наиболее короткими маршрутами между двумя коммутаторами доступа S4 и S5. Возможности настройки топологии сети с помощью протокола PVST продемонстрированы на рисунке 14в. Каналы связи между коммутаторами были распределены на две виртуальные локальные сети, что с одной стороны

позволило использовать все связи в топологии, с а другой стороны появилась возможность балансировки распределения на каналах за счет разделения трафика на два VLAN.

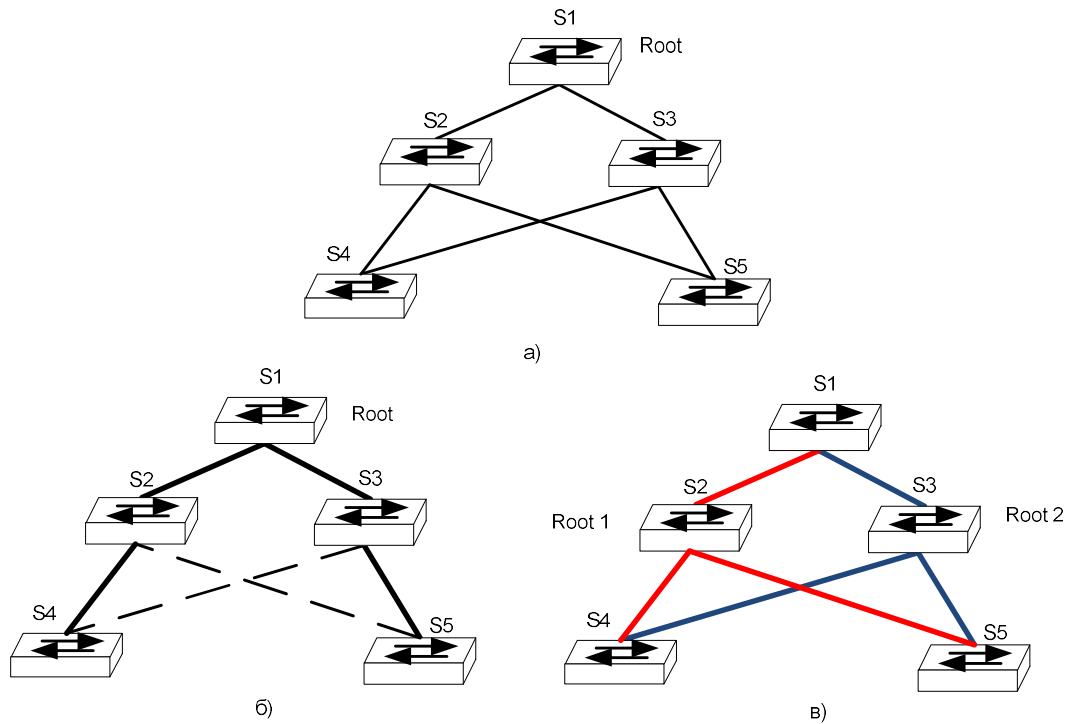


Рисунок 14 – Топология сети. а) топология сети б) активная топология протокола STP в) топология протокола PVST

Этот способ оптимизации передачи сетевого трафика не является универсальным. При большом количестве пользовательских сетей и коммутаторов становится достаточно трудоемким и сложным в планировании топологии, что делает его не лучшим вариантом оптимизации.

1.4.2 SSS

Работа иранских ученых Farhad Faghani и Ghasem Mirjalily направлена на улучшение работы протоколов STP и RSTP. Они предложили два разработанных метода модернизации существующих протоколов.

Первый из них – shortcut switching strategy (метод сокращенных коммутаций). Как известно, протокол STP блокирует некоторые каналы связи в топологии сети. Коммутаторы, которые являются концевыми вершинами дерева топологии и связаны прямыми соединениями, могут потерять эту связь из-за работы STP. Они будут вынуждены обмениваться сообщениями через более длинные маршруты, построенные протоколом, и нагружать своим трафиком те каналы связи, которые могли бы быть свободны.

В процессе изучения топологии сети, коммутатор запоминает адреса не только соседей по назначенным или корневым портам, но и соседей по заблокированным портам. Эти адреса они получают из принятых BPDU. И записывают их в таблицу Blocked Neighbor Nodes (заблокированные соседние узлы), которые в дальнейшем могут быть использованы для прямой пересылки через заблокированный канал.

Метод сокращенных коммутаций позволяет использовать заблокированные каналы в том случае, если физический адрес узла-назначения совпадает с адресом соседнего коммутатора, который расположен на другом конце заблокированного канала.

Порядок работы коммутатора, после принятия кадра выглядит следующим образом:

Если адрес узла-назначения находится в списке Blocked Neighbor Nodes, кадр направляется напрямую к адресату через заблокированный порт;

Если адреса нет в этом списке, кадр обрабатывается согласно традиционному дереву STP.

Пример передачи данных с использованием этого метода можно увидеть на рисунке 15.

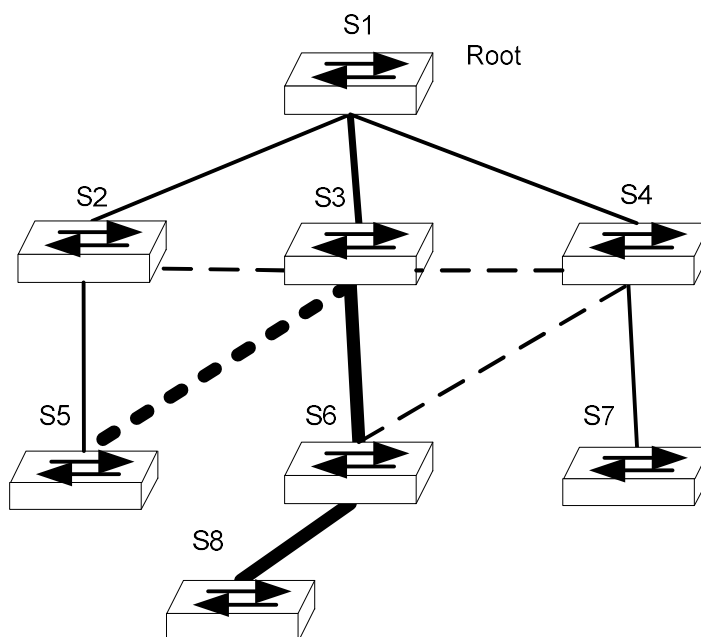


Рисунок 15 – Передача кадра с использованием механизма SSS

На рисунке изображена топология сети с корневым коммутатором S1. Все каналы связи между коммутаторами имеют одинаковую пропускную способность. Пунктирными линиями выделены связи, заблокированные протоколом STP для предотвращения петель. Необходимо передать кадр от коммутатора S8 к коммутатору S5.

Как видно из топологии, при использовании стандартной передачи кадра в протоколе STP, его маршрут будет проложен через коммутаторы S8, S6, S3, S1, S2, S5. Однако, этот маршрут является не самым коротким между конечными точками пути кадра.

1.4.3 IFS

Isolated forwarding strategy – метод изолированных переадресаций – подразумевает под собой дополненный алгоритм работы метода SSS. В этом случае сначала строится топология согласно стандартному протоколу STP. Далее

эта топология разделяется на несколько поддеревьев. Количество поддеревьев равно количеству назначенных портов корневого коммутатора. Каждое поддерево имеет свой номер, который равен идентификатору порта корневого коммутатора, к которому подключено это поддерево. Каждый коммутатор находится в поддереве, за исключением корневого коммутатора, который связан со всеми из них. Также каждый из них должен знать, к какому именно поддереву он относится, поэтому в кадрах BPDU, которые рассылает корневой коммутатор, появилось поле для указания номера поддерева.

В технологии IFS, каждый коммутатор имеет две таблицы коммутации: обычную таблицу STP и таблицу заблокированных соседей (как в методе SSS). Во всех таблицах коммутаций добавлены поля, обозначающие принадлежность узлов к какому-либо из поддеревьев. Помимо этого, изменен формат Ethernet-кадра, в нем появилось два новых поля: номер поддерева узла-источника и номер поддерева узла-назначения. Во время передачи кадров, коммутатор запоминает не только адрес источника и записывает его в таблицу коммутации, но также и номер поддерева, к которому этот источник принадлежит. Если коммутатор не знает номер поддерева узла назначения, такой кадр будет иметь идентификатор поддерева UNKNOWN, то есть неизвестный.

Когда коммутатор получает кадр, он проверяет адрес получателя в своем списке заблокированных соседей, если там находится такой адрес, кадр направляется напрямую через заблокированный порт. В случае его отсутствия коммутатор действует по следующему сценарию:

- если номер поддерева адресата неизвестен или соответствует поддереву, к которому принадлежит коммутатор, кадр отправляется согласно обычному дереву STP;

- если номер поддерева известен, коммутатор просматривает таблицу заблокированных соседей, в случае нахождения там записи об искомом поддереве, коммутатор направляет кадр через заблокированный порт, к

которому присоединен коммутатор из поддерева узла назначения. Если таких портов несколько, выбирается тот, который встретился в таблице последним;

– если в списке заблокированных соседей нет узла с искомым номером поддерева, кадр отправляется согласно обычному дереву STP.

Таким образом, алгоритм IFS старается перенаправить кадр в нужное поддерево при первой же возможности, что значительно снижает нагрузку с каналов, расположенных близко к корневому коммутатору. Пример передачи кадра показан на рисунке 16.

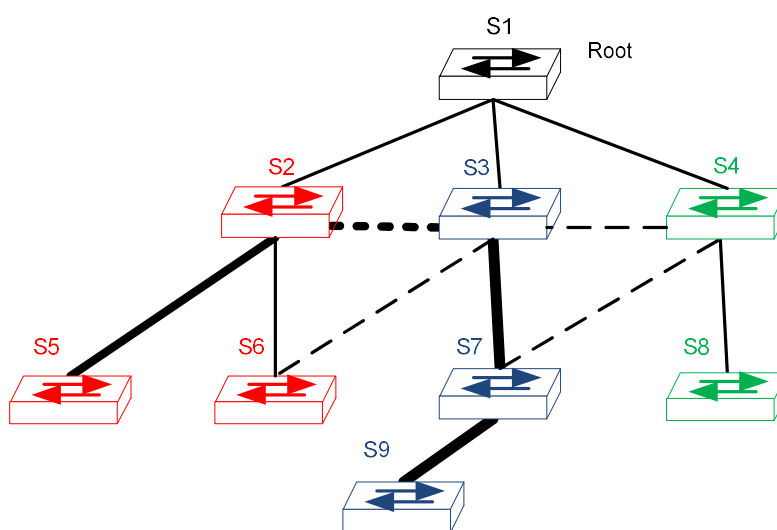


Рисунок 16 – Передача кадра с использованием механизма IFS

2 Протокол множества деревьев

Принимая во внимания недостатки и проблемы в работе протокола STP, создание множества логических топологий связанных деревьев на базе одной физической топологии позволит использовать максимально возможное количество каналов связей между коммутаторами, тем самым распределяя передачу трафика равномерно по всей топологии.

Метод, который разрабатывается в рамках данной работы, заключается в том, что на канальном уровне строится множество остовных деревьев для каждого коммутатора, вместо одного общего дерева с коммутатором-корнем, который выбирается один на весь сегмент. В таком случае, в процессе передачи кадра участвуют только наиболее оптимальные маршруты, а вследствие децентрализованности управления процессом построений деревьев увеличивается время сходимости сети при сбоях.

2.1 Область применения

Маршрутизация трафика на канальном уровне хорошо применима для высокоскоростных сетей провайдера, который связывает отдельные сети компаний. Для клиентов предоставляется один или несколько портов в выходных маршрутизаторах, а внутренняя топология сети провайдера строится на высокоскоростных коммутаторах. Потери некоторого времени на запаковку и распаковку кадров являются целесообразными, так как за счет этого уменьшается время передачи пакетов с использованием лучших маршрутов, а также снизится нагрузка на некоторые коммутаторы и каналы связи, по сравнению с их работой в обычном протоколе STP.

В целом такая система выглядит как один виртуальный коммутатор, который имеет точки входа и выхода для клиентов и состоит из множества физических коммутаторов и связей между ними (Рисунок 17).

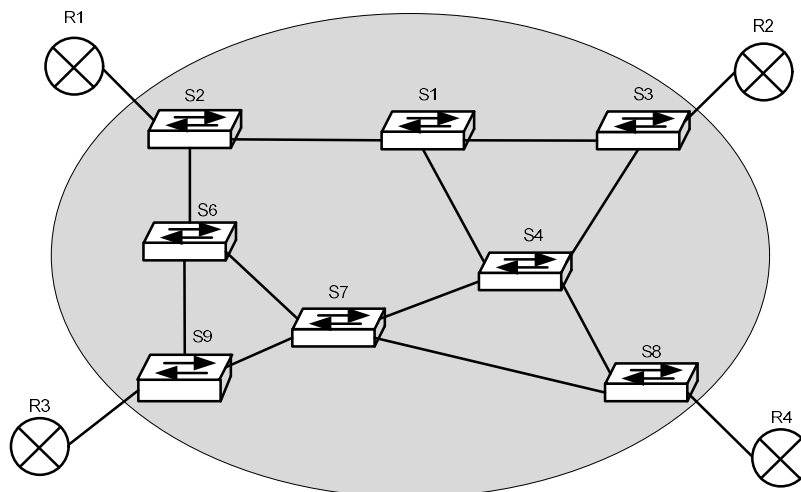


Рисунок 17 – Топология сети

2.2 Основные понятия

- Tree ID – идентификатор дерева внутри сегмента сети. Tree ID совпадает с Bridge ID коммутатора, который является корневым в рамках этого дерева;
- Bridge ID – идентификатор коммутатора в сети. Используется для адресации кадров в протоколе множественных деревьев;
- Cost – стоимость локального порта коммутатора. Используется только в пределах коммутатора;
- Pathcost – стоимость пути до корневого коммутатора в дереве;
- Root port – магистральный порт, который направлен в сторону корневого коммутатора в определенном дереве и принимает от него кадры;
- Designated port – магистральный порт, который используется для дальнейшей пересылки кадров по определенному дереву;
- Alternative port – резервный магистральный порт для определенного дерева, который принимает и посылает служебные BPDU, но не пересылает кадры данных;
- Client port – немагистральный порт, к которому подключено клиентское оборудование;

- Main port – магистральный порт, участвующий в процессе передачи кадра;
- Client MAC table – таблица MAC-адресов устройств, подключенных к немагистральным портам коммутатора;
- Tree ID table – таблица, которая хранит записи обо всех Tree ID в сегменте, и о ролях всех магистральных портов конкретного коммутатора в них.

2.3 Принцип построения деревьев

Каждый из коммутаторов при включении в сегмент сети рассылает через все свои магистральные интерфейсы кадры построения дерева. После того, как такой кадр принял коммутатор на какой-либо из своих магистральных портов, он обрабатывает кадр, принимает решение о роли порта в дереве с указанным в кадре идентификатором и отправляет ответный кадр «Готов».

В случае если коммутатор имеет свою таблицу клиентских MAC-адресов, он также отправляет её в ответ. Таким образом, корневой коммутатор своего дерева может привязывать различные таблицы клиентских MAC-адресов к своим разным интерфейсам, что позволяет направлять кадры по маршруту, а также связывать входные данные по MAC-адресам с идентификаторами, которые используются в протоколе при адресации (Рисунок 18).

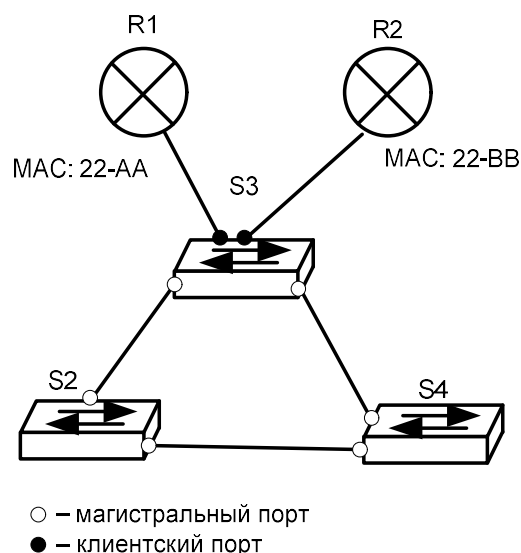


Рисунок 18 – Топология сети

После того, как коммутатор ответил сообщением «Готов», он пересылает сообщение о конфигурации на другие свои магистральные порты, тем самым продолжая процесс формирования дерева. Также необходимо отметить, что при получении кадра конфигурации с построением дерева для определенного Bridge ID, коммутатор запоминает интерфейс, с которого пришел данный кадр. Это необходимо для того, чтобы в случае пересылки сообщения «Готов» от нижестоящих коммутаторов, была возможность направить это сообщение через нужный интерфейс

Каждый порт коммутатора имеет свою метрику, она зависит от скорости того или иного интерфейса. Чем выше скорость интерфейса, тем меньше его метрика. Тем самым топология коммутаторов представляет собой взвешенный граф, вес ребер которого представляют собой метрики каналов. Каждый конфигурационный кадр хранит в себе полную метрику маршрута до коммутатора, который стал источником этого кадра. При получении одного или нескольких таких кадров, коммутатор имеет возможность сравнить метрику маршрута до одного и того же коммутатора через различные интерфейсы (Рисунок 19).

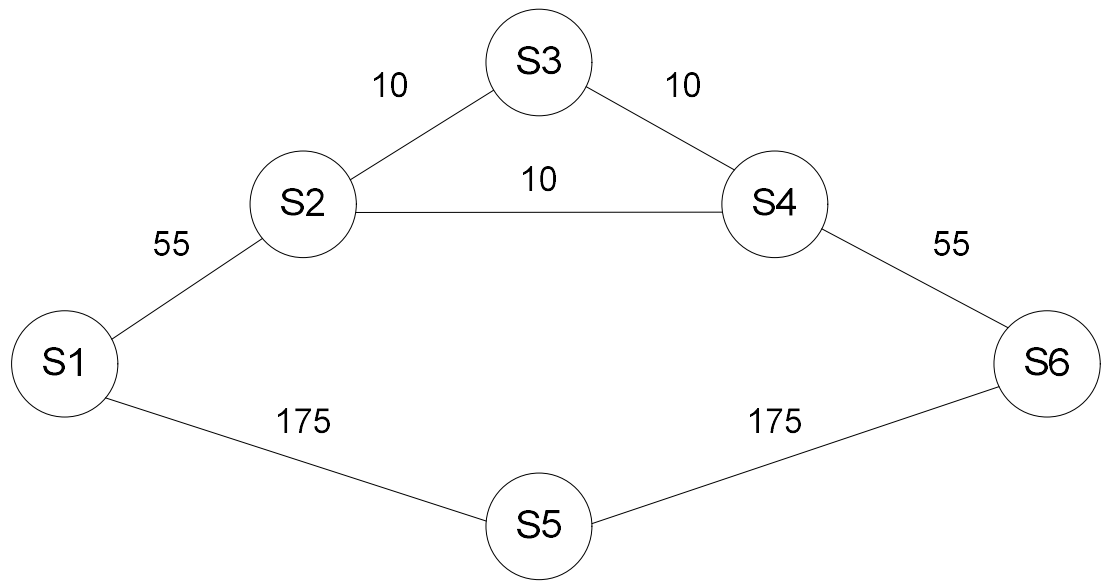


Рисунок 19 – Топология сети в виде взвешенного графа

2.4 Состояния и роли портов

Порты могут находиться в двух физических состояниях как на рисунке 20:

- down – порт выключен;
- active – порт включен, принимает и отправляет BPDU, принимает и отправляет данные в рамках конкретного дерева.

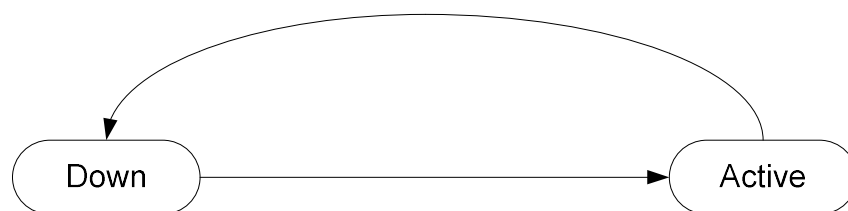


Рисунок 20 – Состояния портов

Порт может иметь одну из нескольких состояний в рамках какого-либо дерева, эти состояния являются логическими, так как все они присущи только включенному интерфейсу как на рисунке 21:

– Root – порт является корневым портом коммутатора для определенного дерева, то есть он будет принимать пакеты, направленные от коммутатора, который является корневым в экземпляре дерева;

– Designated – порт будет выходным портом коммутатора для определенного экземпляра дерева;

– Alternative – порт является альтернативным для определенного экземпляра дерева, в случае потери Root-порта коммутатор будет использовать альтернативный канал как Root.

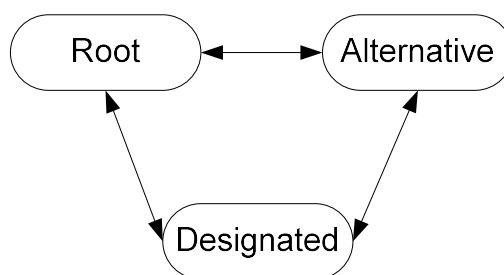


Рисунок 21 – Роли портов

2.5 База данных коммутатора

Для успешной работы коммутатора, ему необходимо хранить информацию о состоянии каналов связи, выходных адресах интерфейсов для передачи кадров на другие коммутаторы, а также иметь таблицы коммутации для коммутирования входящих кадров в таблице 5.

Вся эта информация располагается в таблицах четырех видов, которые хранятся в памяти коммутатора в таблице 6.

Первый из них – таблица ролей портов, она существует в единственном экземпляре и описывает все магистральные порты коммутатора, все известные ему Tree ID, роли, которые имеют эти порты для всех Tree ID, а также метрики магистральных каналов в таблице 7.

Второй вид – таблица привязки клиентских таблиц к портам, так же является уникальной в рамках коммутатора, в ней хранятся соответствия

клиентских таблиц, которые присылают коммутаторы вместе с кадром «Готовность».

Третий вид таблиц – таблицы клиентских MAC-адресов. Эти таблицы состоят из списка MAC-адресов устройств, которые подключены к клиентским портам коммутаторов, которые прислали ответный кадр «Готовность».

Помимо этого, коммутатор должен хранить собственную таблицу коммутации для работы с устройствами на клиентских портах в таблице 8.

Кроме таблиц, коммутатор хранит данные о метриках собственных магистральных интерфейсов, для того, чтобы иметь возможность пересчитывать метрику пути до удаленного коммутатора при распространении конфигурационной информации в таблице 9.

На примере коммутатора S2 на рисунке продемонстрированы примеры данных таблиц (Рисунок 22).

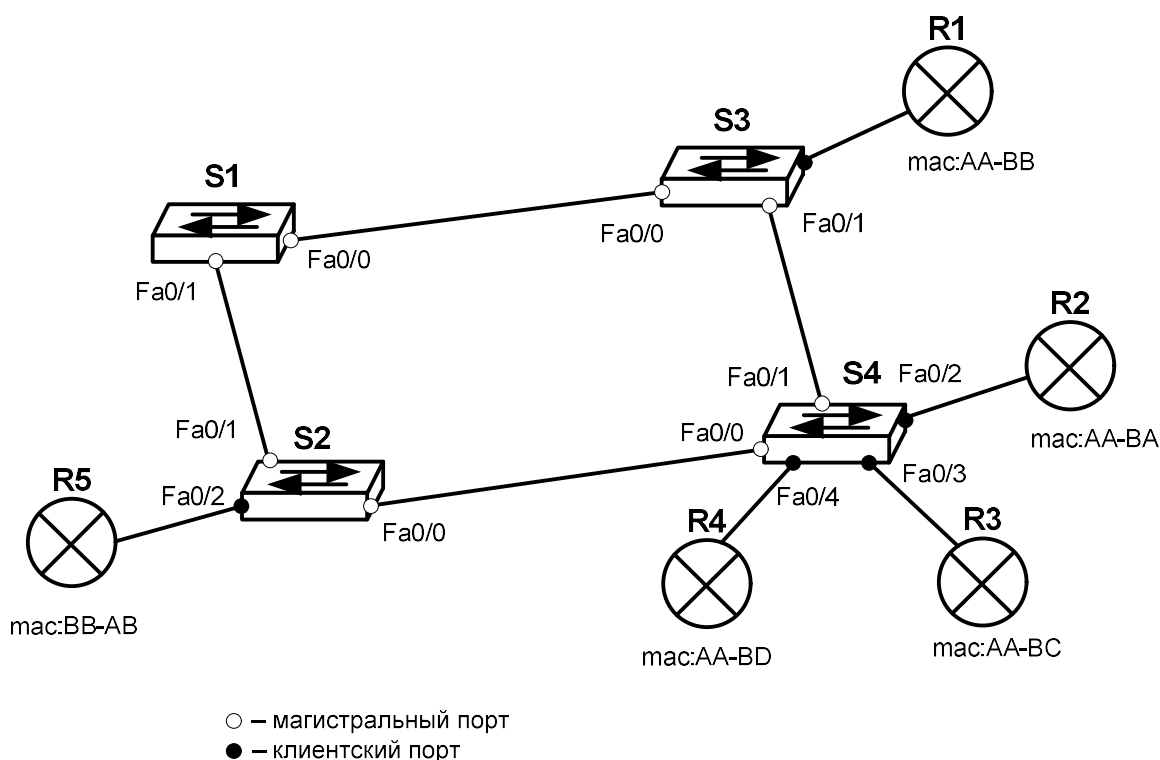


Рисунок 22 – Топология сети

Таблица 5 – Таблица ролей портов

Tree Id	FastEthernet 0/0	Metric	FastEthernet 0/1	Metric
S1	Designated	-	Root	10
S3	Alternative	20	Root	20
S4	Root	10	Designated	-

Таблица 6 – Таблица собственных клиентских MAC-адресов

Tree ID	MAC-addresses
S3	AA-BB

Таблица 7 – Таблица клиентских MAC-адресов

Tree ID	MAC-addresses
S4	AA-BA
	AA-BC
	AA-BD

Таблица 8 – Таблица привязки

Tree ID	Outbound Interface
S3	FastEthernet 0/1
S4	FastEthernet 0/0

Таблица 9 – Таблица коммутации

MAC-address	Outbound Interface
BB-AB	FastEthernet 0/2

2.6 Формат кадров

Для обеспечения работы протокола был изменен стандартный кадр Ethernet (Рисунок 23). Подобно кадру стандарта 802.1Q, все изменения кадра вносились после полей преамбулы, MAC-адресов получателя и отправителя. Это необходимо для того, чтобы оборудование, которое предназначено только для работы с обычными кадрами Ethernet и не умеет работать с нестандартными заголовками, смогло перенаправить кадр дальше по маршруту.

Ethernet-кадр

Преамбула	MAC-адрес получателя	MAC-адрес отправителя	EtherType	Данные	Контрольная сумма
8 байт	6 байт	6 байт	2 байта	46-1500 байт	4 байта

Кадр разрабатываемого протокола

Преамбула	MAC-адрес получателя	MAC-адрес отправителя	Bridge ID/Tree ID	Тип кадра	EtherType	Данные	Контрольная сумма
8 байт	6 байт	6 байт	6 байт	1 байт	2 байта	46-1493 байта	4 байта

Рисунок 23 – Отличия кадров

К заголовкам было добавлено два поля: Bridge ID/Tree ID и тип кадра. Тип кадра указывает на его предназначение. Кадры могут быть следующих типов:

- кадр конфигурации;
- кадр «потеря Root»;
- кадр «Готов»;
- кадр «Канал загружен»;
- кадр «Предложение Root-порта»;
- кадр «Изменение состояния канала»;
- кадр «Установка отношений Designated»;
- кадр «Установка отношений Alternative»;
- кадр данных.

В зависимости от типа кадра можно по-разному интерпретировать значение поля Bridge ID/Tree ID. Если тип пришедшего кадра – данные, то есть неслужебная информация, под значением поля будет пониматься адрес получателя данного кадра в рамках передачи его по магистральным каналам. На основе этой информации будет выбран выходной интерфейс для дальнейшей передачи кадра на магистральный или клиентский интерфейс.

Таким образом, после добавления в заголовок кадра дополнительных полей, размер заголовка увеличился с 26 до 33 байт. Соответственно,

максимально возможный размер данных, передаваемых в кадре, уменьшился на 7 байт. В процентном соотношении увеличение минимального размера кадра будет на 9,7%, а максимальный размер кадра останется прежним. Говоря о передаваемых кадром данных, нужно отметить, что максимально возможный их размер уменьшился на 0,5 %.

2.7 Алгоритмы работы коммутатора

Определены следующие алгоритмы работы коммутаторов:

- начало построения своего дерева;
- предложение об установке отношений;
- действия при потере канала;
- действия при подключении канала;
- действия при необходимости отправить данные;
- действия при подключении клиентского оборудования;
- обработка кадра «Конфигурация дерева»;
- обработка кадра «Готовность»;
- обработка кадра «Предложение отношений»;
- обработка кадра «Изменена роль порта»;
- обработка кадра «Потеряна root-связь»;
- обработка кадра «Уведомление о состоянии канала»;
- обработка кадра данных;
- блок-схемы алгоритмов представлены в приложениях А-Л.

2.8 Борьба с broadcast-штормами

Как и в любой сети, в топологии коммутаторов может возникнуть ситуация, когда коммутатору необходимо отправить кадр, но он не будет знать

точный адрес узла назначения. В таком случае кадр будет послан на широковещательный адрес FF:FF:FF:FF:FF:FF и все коммутаторы будут рассылать этот кадр на все свои порты, кроме того, который получил этот кадр. В разрабатываемом протоколе так же используется широковещательный адрес узла-назначения.

При обработке широковещательного кадра, каждый коммутатор будет считывать из заголовка адрес MAC-адрес узла-источника и, исходя из этих данных, отправляет их по какому-либо из TreeID. Таким образом, широковещательный кадр не будет бесконечно распространяться по сегменту, так как будет распространяться только в рамках одного дерева, с TreeID равного BridgeID коммутатора, от которого широковещательный кадр попал в домен.

2.9 Поддержка отношений

Так как при каких-либо изменениях в топологии будут рассылаться кадры уведомления различного типа или кадры конфигурации, периодическая рассылка служебной информации будет излишней. Однако, на практике возможны ситуации, когда кадры уведомления не смогут достичь адресата, поэтому периодически, для проверки работоспособности канала связи коммутатор инициирует рассылку контрольных BPDU.

В случае получения ответа от соседнего коммутатора, канал считается рабочим, если ответа на проверочный BPDU не приходит, канал считается неактивным и порт переходит в состояние down до тех пор, пока связь не восстановится.

3 Разработка проекта

Для проведения моделирования работы сети с использованием обычного протокола STP и разрабатываемого протокола была использована среда моделирования Omnet++, которая позволяет построить модель вычислительной сети, задать параметры передачи данных, пропускной способности каналов. Также программа позволяет получить выходные данные для последующего анализа, после того, как отработает процесс моделирования активности в сети.

3.1 Полученные результаты

На рисунках 24 и 25 показаны пути передачи данных в первой экспериментальной топологии.

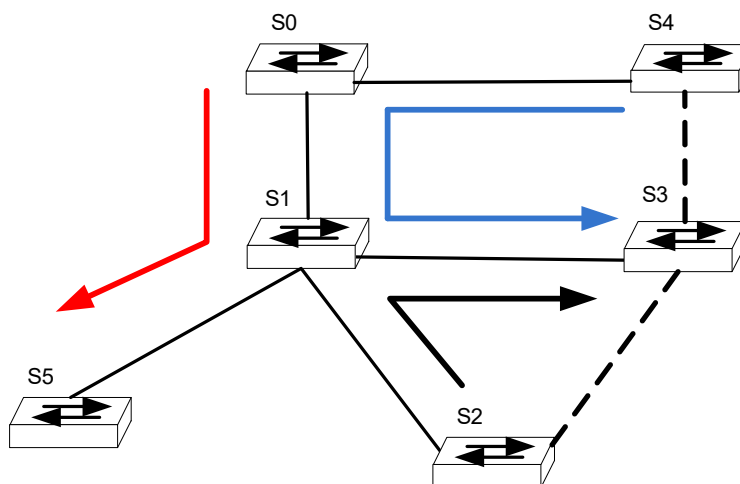


Рисунок 24 – Маршруты передачи в STP

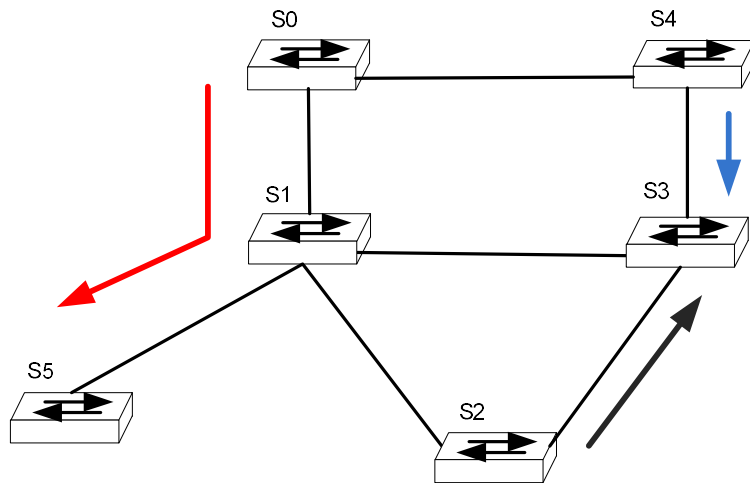


Рисунок 25 – Маршруты передачи в разработанном протоколе

Сравнительные полученные значения очередей на выходах коммутаторов, где пересекаются два и более потоков, а также времени передачи данных в первом эксперименте представлены в таблицах 10 и 11.

Таблица 10 – Максимальный размер очереди

Выходной интерфейс	STP	Разработанный протокол
S	кб	0.1 кб
S	6.1кб	0.1 кб

Таблица 11 – Данные о передаваемой информации

Поток данных	STP	Разработанный протокол
S	2.019 с	2.004 с
S	1.041 с	1.002 с
S	1.022 с	1.002 с

На рисунках 26 и 27 показаны пути передачи данных во второй экспериментальной топологии.

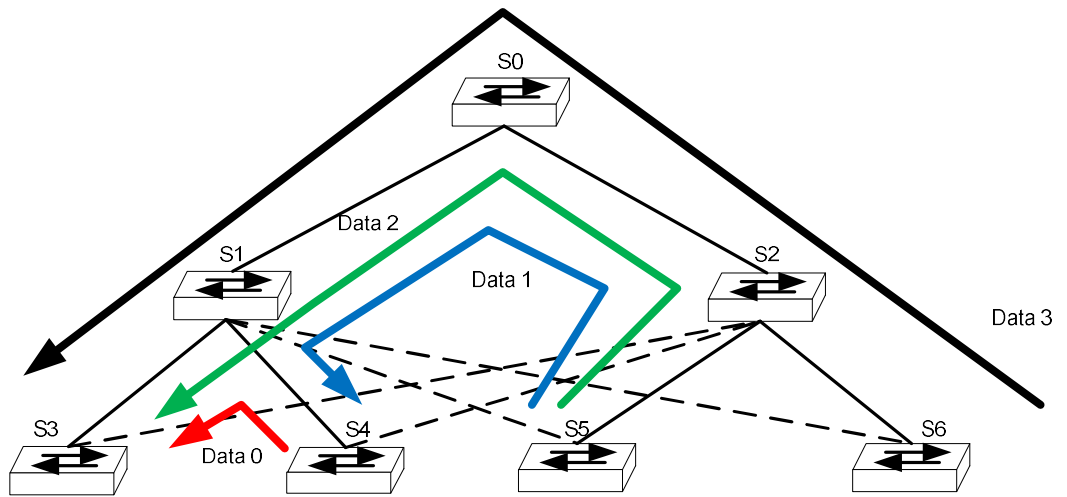


Рисунок 26 – Маршруты передачи в STP

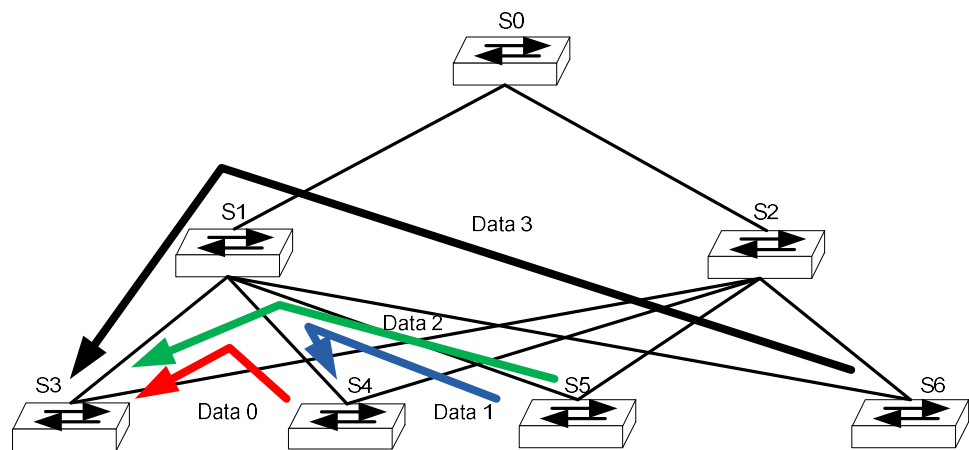


Рисунок 27 – Маршруты передачи в разработанном протоколе

Сравнительные полученные значения очередей на выходах коммутаторов, где пересекаются три и более потоков, а также времени передачи данных в первом эксперименте представлены в таблицах 12 и 13.

Таблица 12 – Максимальный размер очереди

Выходной интерфейс	STP	Разработанный протокол
S	кб	0.1 кб
S	кб	0.1 кб
S	кб	8.6 кб

Таблица 13 – Данные о передаваемой информации

Поток данных	STP	Разработанный протокол
S	4.39 с	534 с
S	264 с	556 с
S	с	с
S	81 с	2.009 с

На рисунках 28 и 29 показаны пути передачи данных в третьей экспериментальной топологии.

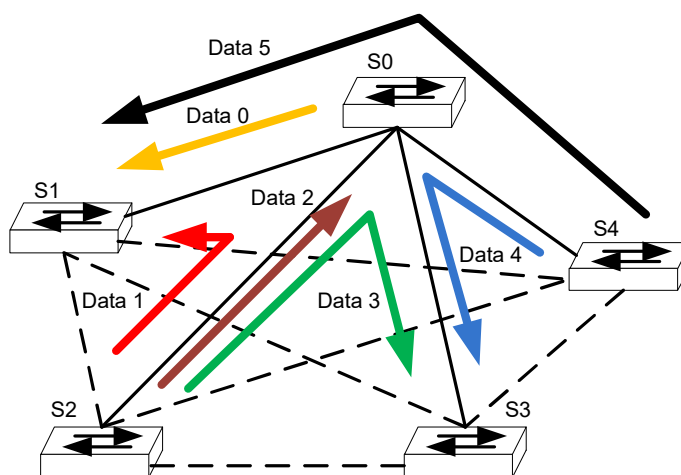


Рисунок 28 – Маршруты передачи в STP

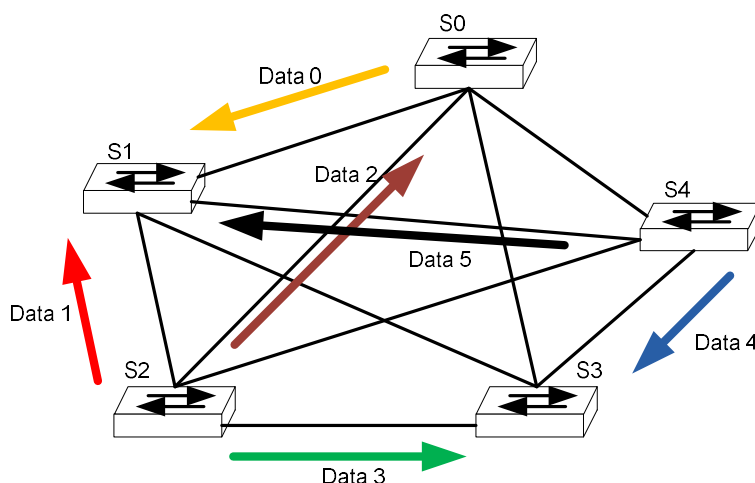


Рисунок 29 – Маршруты передачи в разработанном протоколе

Сравнительные полученные значения очередей на выходах коммутаторов, где пересекаются три и более потоков, а также времени передачи данных в первом эксперименте представлены в таблицах 14 и 15.

Таблица 14 – Максимальный размер очереди

Выходной интерфейс	STP	Разработанный протокол
S	кб	0.1 кб
S	кб	0.1 кб
S	кб	0.1 кб

Таблица 15 – Данные о передаваемой информации

Поток данных	STP	Разработанный протокол
S	7.895 с	6.013 с
S	7.836 с	7.015 с
S	.39 с	4.009 с
S	3.199 с	3.007 с
S	5.022 с	5.011 с
S	5.784 с	5.011 с

На рисунках 30 и 31 показаны пути передачи данных в четвертой экспериментальной топологии.

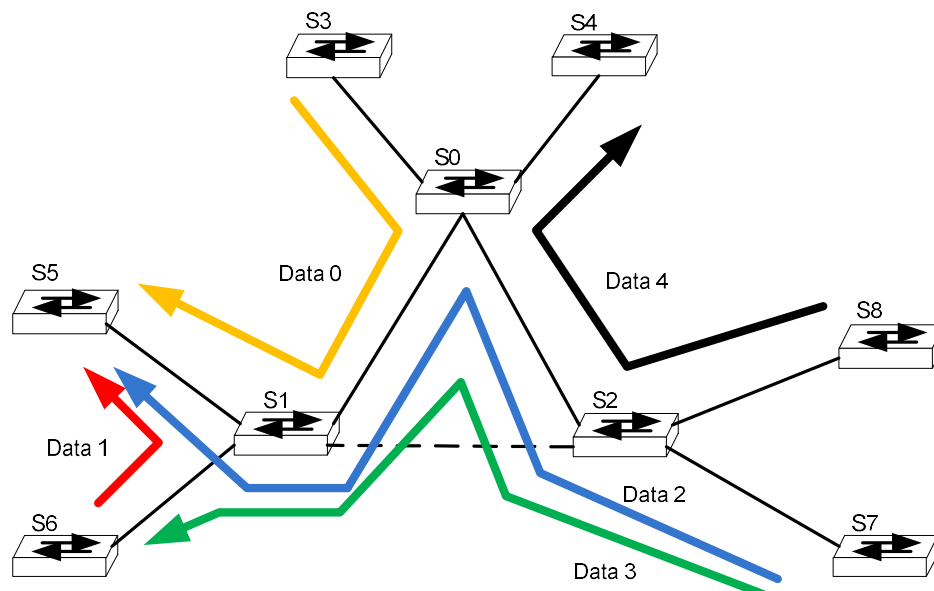


Рисунок 30 – Маршруты передачи в STP

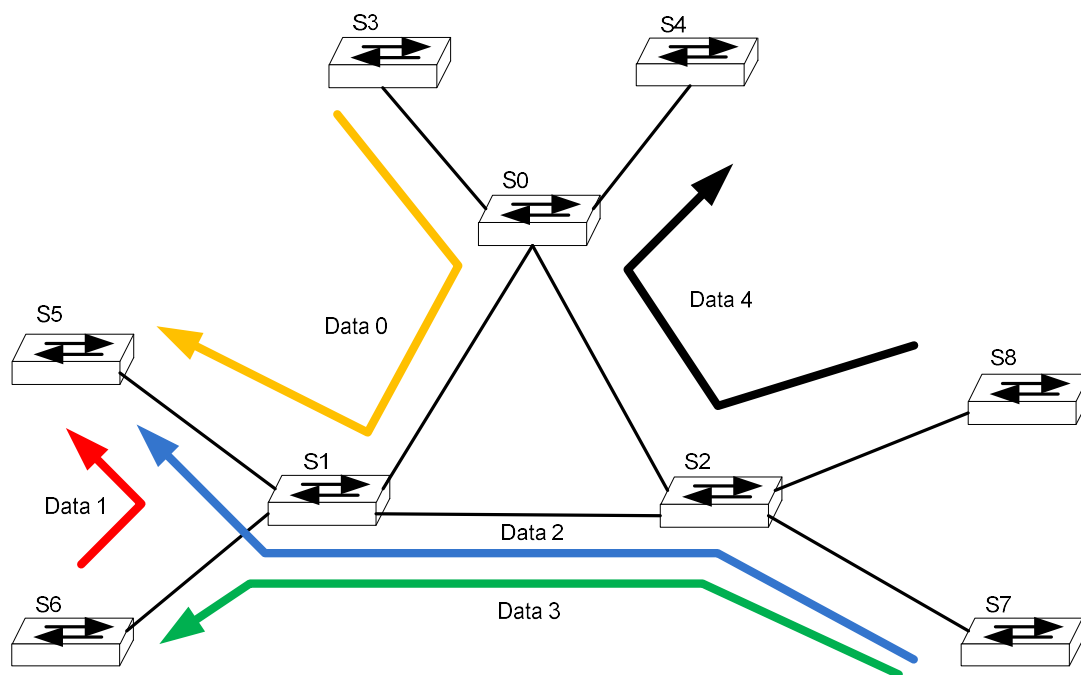


Рисунок 31 – Маршруты передачи в разработанном протоколе

Сравнительные полученные значения очередей на выходах коммутаторов, где пересекаются три и более потоков, а также времени передачи данных в первом эксперименте представлены в таблицах 16 и 17.

Таблица 16 – Максимальный размер очереди

Выходной интерфейс	STP	Разработанный протокол
S0-S1	72.4 кб	0.1 кб
S	30.2 кб	44.6 кб
S2-S0	130.1 кб	0.1 кб

Таблица 17 – Данные о передаваемой информации

Поток данных	STP	Разработанный протокол
S	6.048 с	5.501 с
S	5.172 с	5.079 с
S	5.172 с	4.317 с
S	6.279 с	6.013 с
S	4.945 с	4.009 с

Из полученных результатов можно увидеть, что время передачи данных при использовании разработанного протокола в большинстве случаев уменьшилось. В зависимости от количества связей в топологии эффективность протокола различна. Например, в третьем эксперименте, в случае полносвязной топологии были получены наилучшие результаты, так как любые потоки данных между узлами не пересекаются. В частных случаях, некоторые потоки данных в разработанном протоколе передаются медленнее, чем в протоколе STP, однако общее время передачи данных по всей совокупности потоков при этом уменьшается.

ЗАКЛЮЧЕНИЕ

В ходе написания работы была проанализирована область сетей передачи данных, выявлены направления, которые имеют свои недостатки – в данном случае передача кадров на канальном уровне. Были поставлены задачи, решение которых позволит избавиться от выявленных недостатков. Разработанный протокол канального уровня, который позволит передавать кадры более эффективно. В результате моделирования, метод должен снизить нагрузку на корневой узел и околокорневые каналы, распределив ее по всей площади сети. Общая средняя скорость потока данных, в следствии меньшей нагрузки на определенные точки сети, должна возрасти.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Российская государственная библиотека : официальный сайт. – Москва, 1999– . – URL: <http://www.rsl.ru> (дата обращения: 26.06.2019).
- 2 Введение в Spanning Tree протокол // Bag traq : официальный сайт. – 1999–. – URL: <http://bugtraq.ru/library/books/stp/chapter01/> (дата обращения: 17.04.2023).
- 3 Вычислительные сети. Теория и практика // BCNV : официальный сайт. – 2003– . – URL: <http://network-journal.mpei.ac.ru/cgi-bin/main.pl?l=ru&n=1&pa=2&ar=1> (дата обращения: 21.05.2023).
- 4 Закольцованные сети, или зачем нам STP // Habr : официальный сайт. – 2003– . – URL: <http://habrahabr.ru/post/129559> (дата обращения: 21.05.2023).
- 5 Маршрутизация через несколько каналов/провайдеров // Opennet : официальный сайт. – 2001– . – URL: <http://www.opennet.ru/docs/RUS/LARTC/x348.html> (дата обращения: 27.05.2023).
- 6 Принцип работы протоколов STP/RSTP, функции защиты корня (root guard) // НСК Коммуникации Сибири : официальный сайт. – 2023 – URL: <http://nsc-com.com/?page=155> (дата обращения: 03.06.2023).
- 7 Сети для самых маленьких. Часть четвертая. STP // Linkmeup : официальный сайт. – 2022 – URL: <http://linkmeup.ru/blog/15.html> (дата обращения: 17.05.2023).
- 8 The Spanning Tree Protocol // Omnetpp : официальный сайт. – 2023 – URL: <https://doc.omnetpp.org/inet/api-current/neddoc/inet.linklayer.ieee8021d.stp.Stp.html> (дата обращения: 19.05.2023).
- 9 Трехуровневая иерархическая модель компании Cisco // Network : официальный сайт. – 1999– . – URL: http://network.xsp.ru/8_5_7.php (дата обращения: 17.06.2023).

10 Программные средства измерения производительности лс ПЭВМ // Программные продукты и системы : официальный сайт. – 2007– . – URL: <http://swsys.ru/index.php?page=article&id=1182> (дата обращения: 04.06.2023).

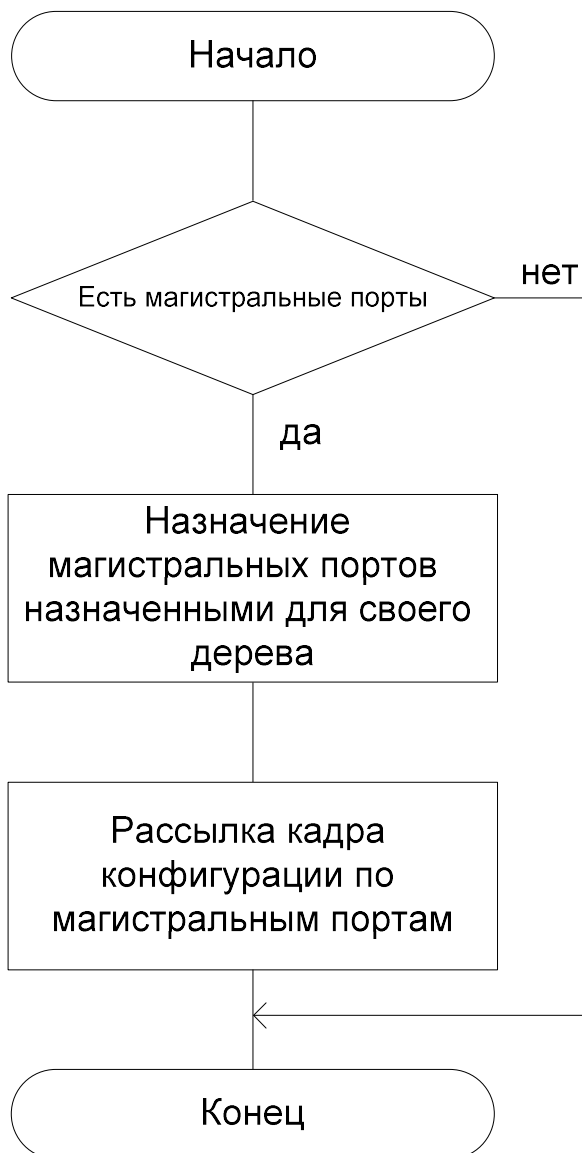
11 Configuration example to migrate Spanning Tree from PVST+ to MST - Cisco Systems // Cisco : официальный сайт. – 2023 – URL: http://www.cisco.com/en/US/products/hw/switches/ps708/products_configuration_example09186a00807b075f.shtml (дата обращения: 11.06.2023).

12 Configuring STP // Cisco : официальный сайт. – 2023 – URL: http://www.cisco.com/en/US/docs/switches/lan/catalyst2960/software/release/12.2_5_5_se/configuration/guide/swstp.html#wp1039614 (дата обращения: 09.06.2023).

13 IEEE 802.1: 802.1Q // Virtual LANs : официальный сайт. – 2023 – URL: <http://www.ieee802.org/1/pages/802.1Q.html> (Дата обращения: 03.05.2023)

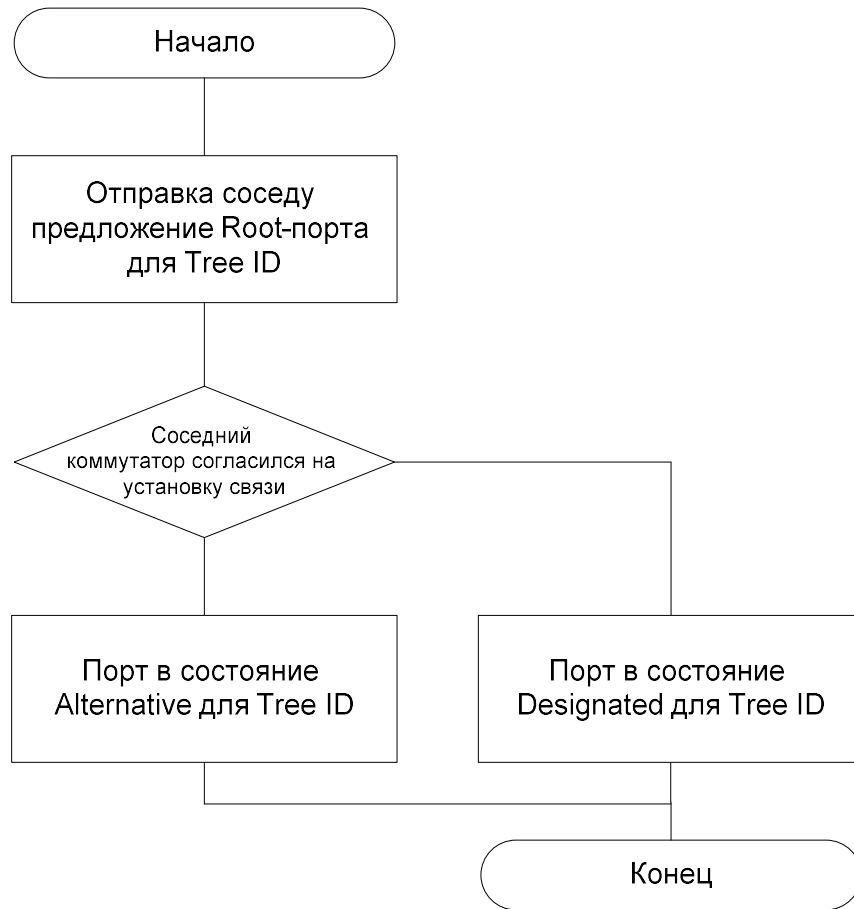
14 Shortcut Switching Strategy in Metro Ethernet networks // ScienceDirect : официальный сайт. – 2023 – URL: <http://www.sciencedirect.com/science/article/pii/S0140366410004792> (Дата обращения: 20.06.2023)

ПРИЛОЖЕНИЕ А
Включение коммутатора



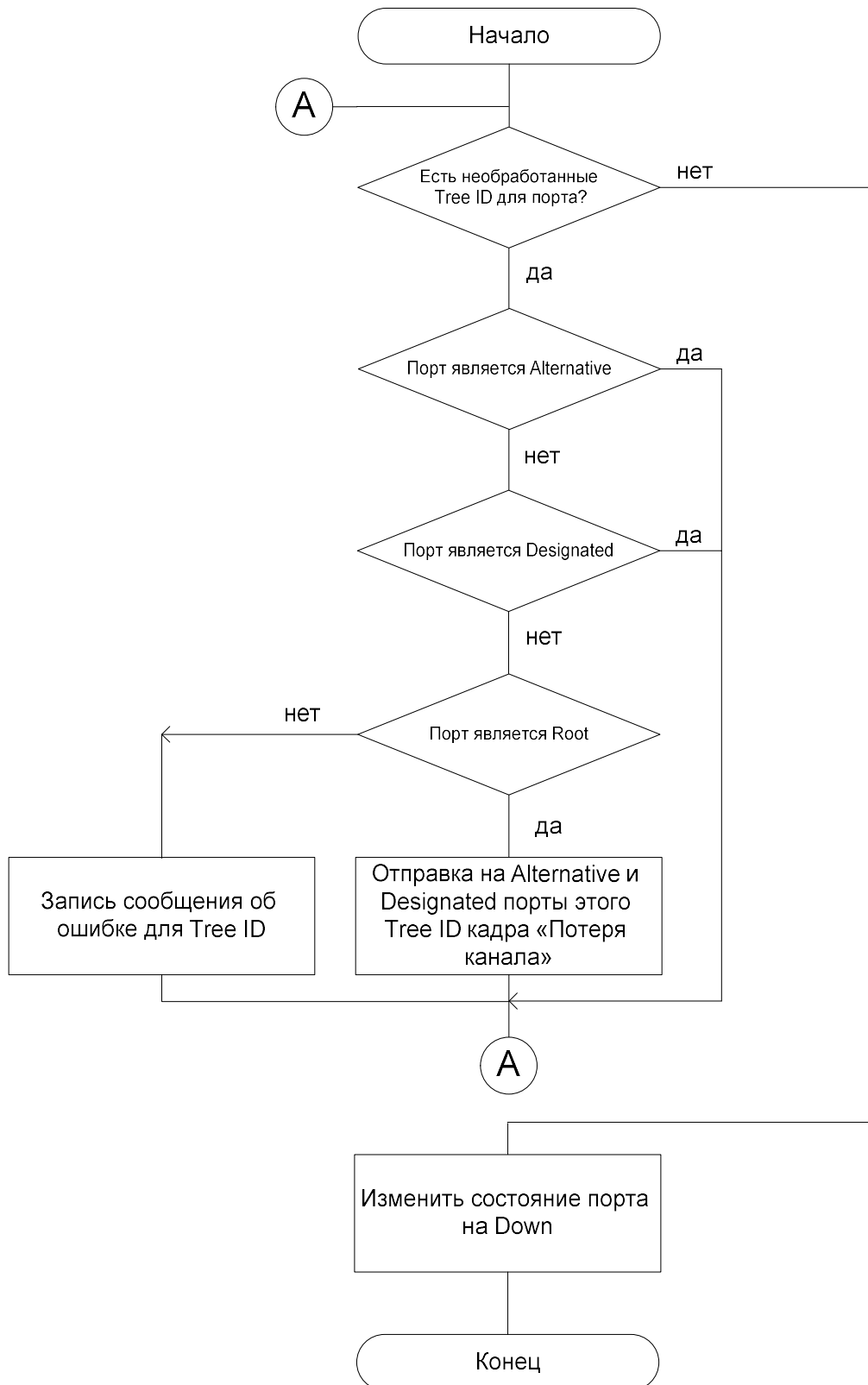
ПРИЛОЖЕНИЕ Б

Предложение об установке связи



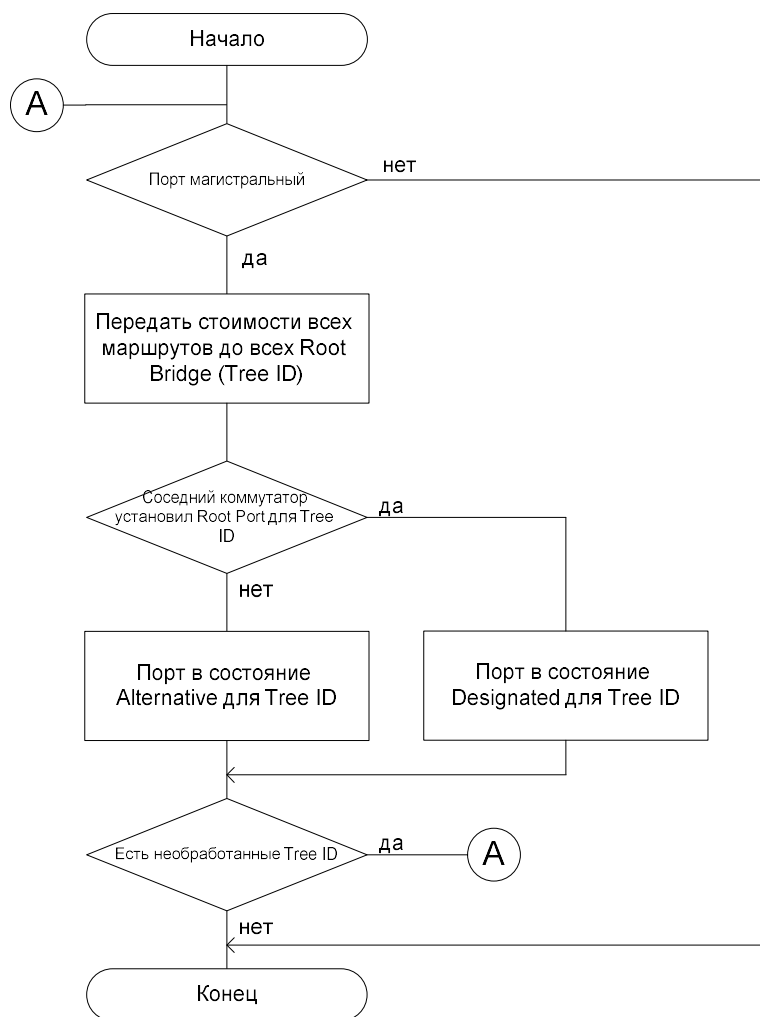
ПРИЛОЖЕНИЕ В

Действия при потере канала связи



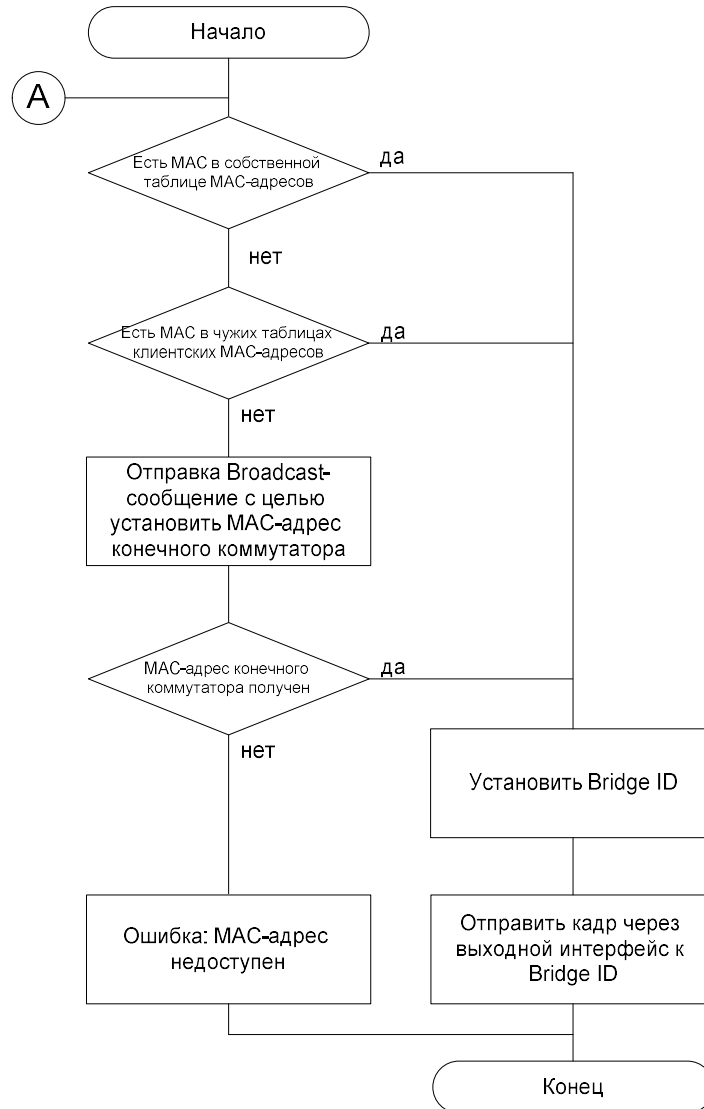
ПРИЛОЖЕНИЕ Г

Действия при подключении канала



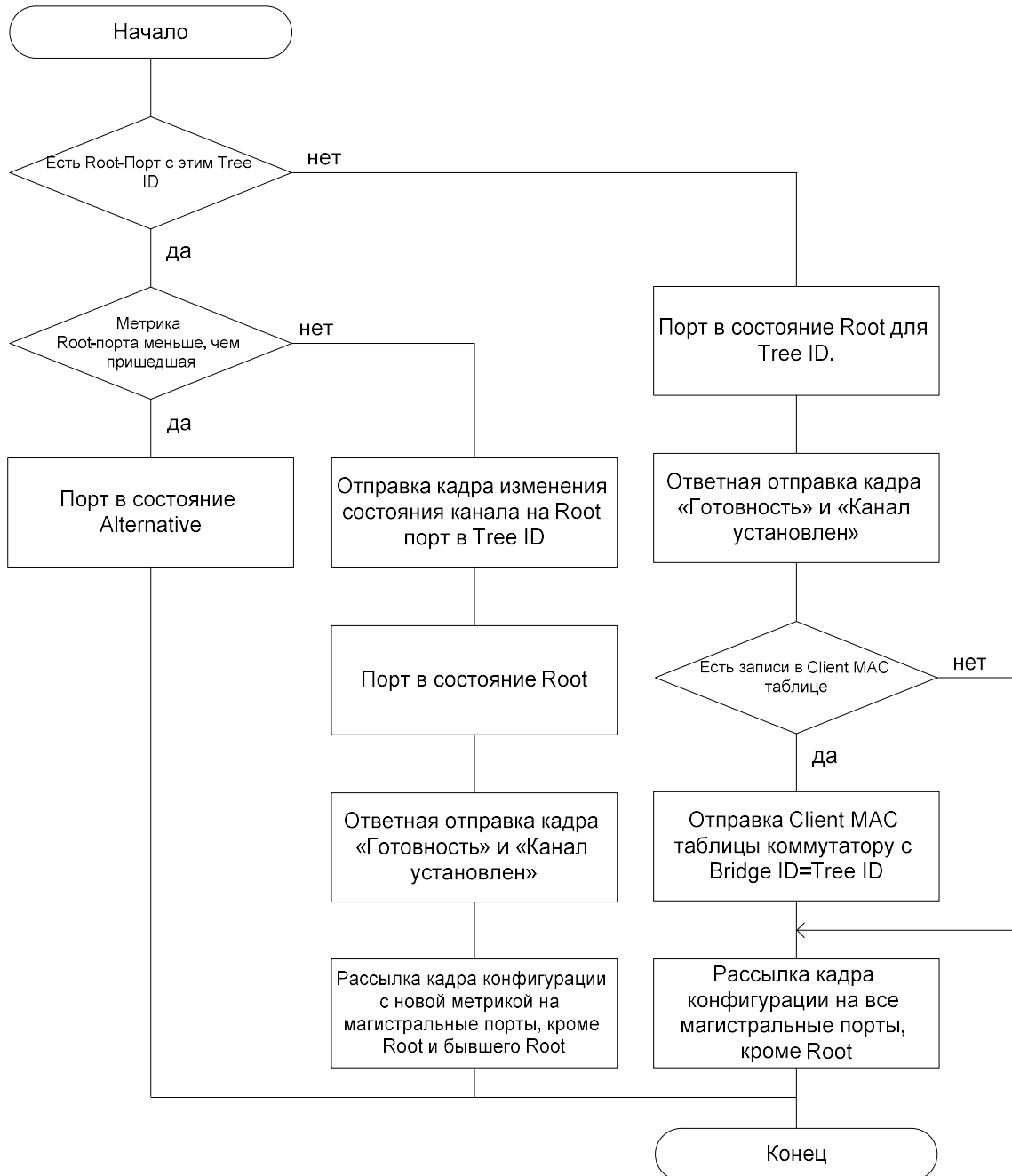
ПРИЛОЖЕНИЕ Д

Действия при необходимости отправки данных



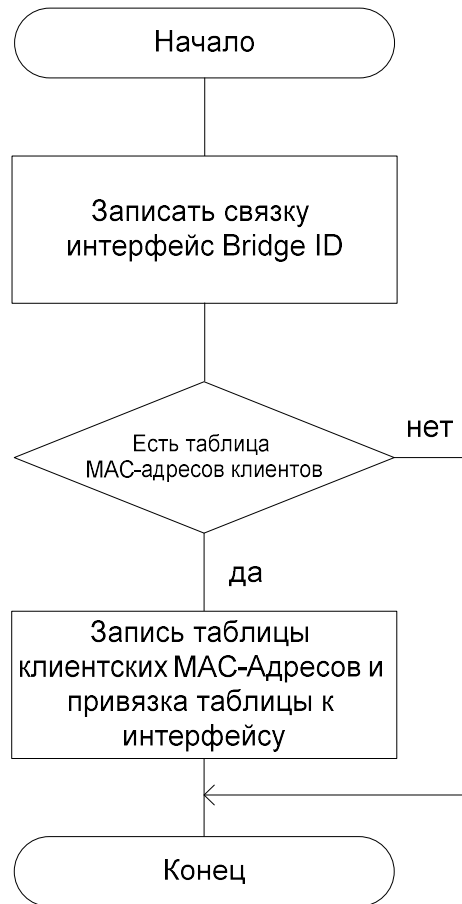
ПРИЛОЖЕНИЕ Е

Обработка кадра «Конфигурация дерева»



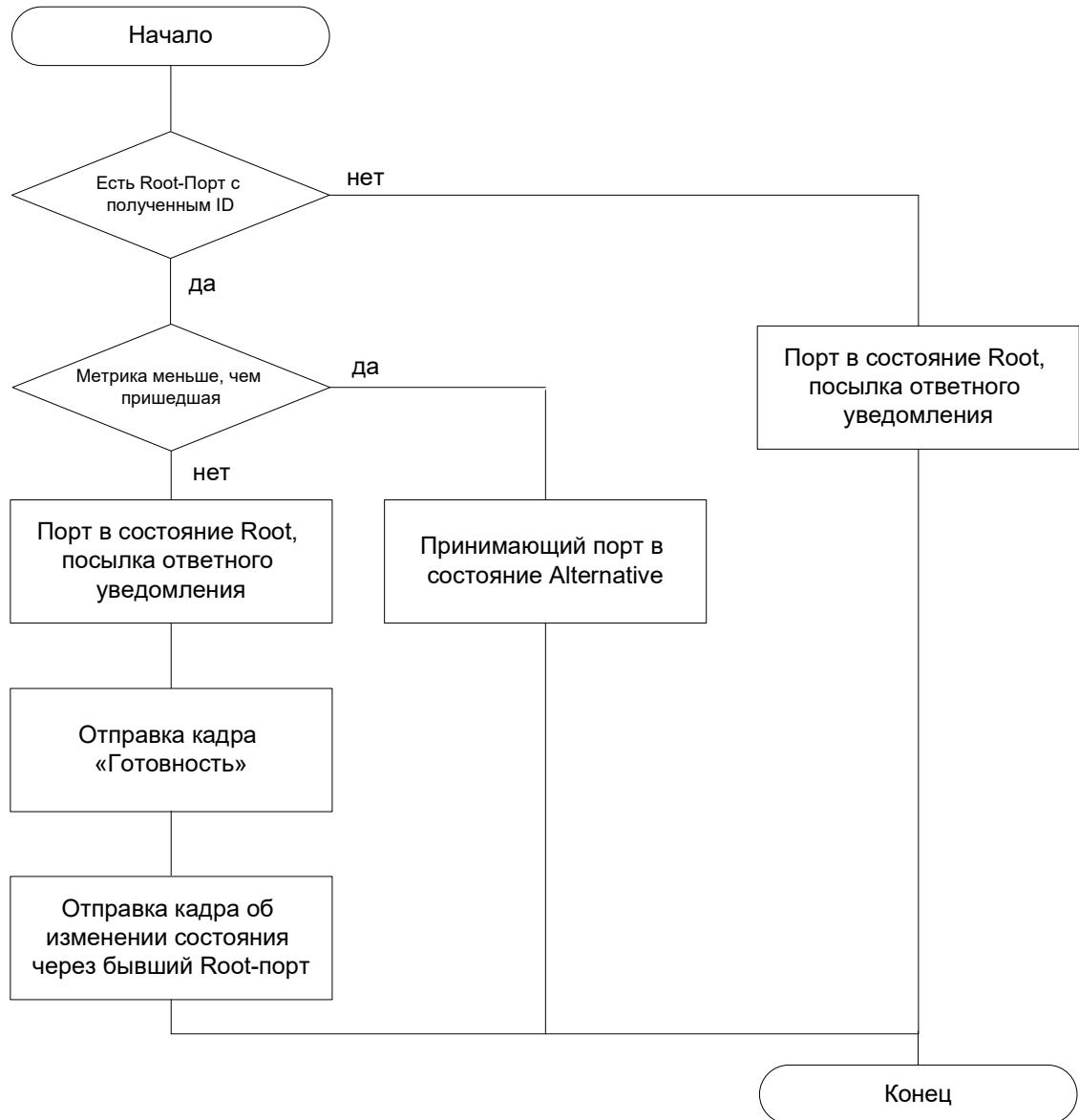
ПРИЛОЖЕНИЕ 3

Обработка кадра «Готовность»



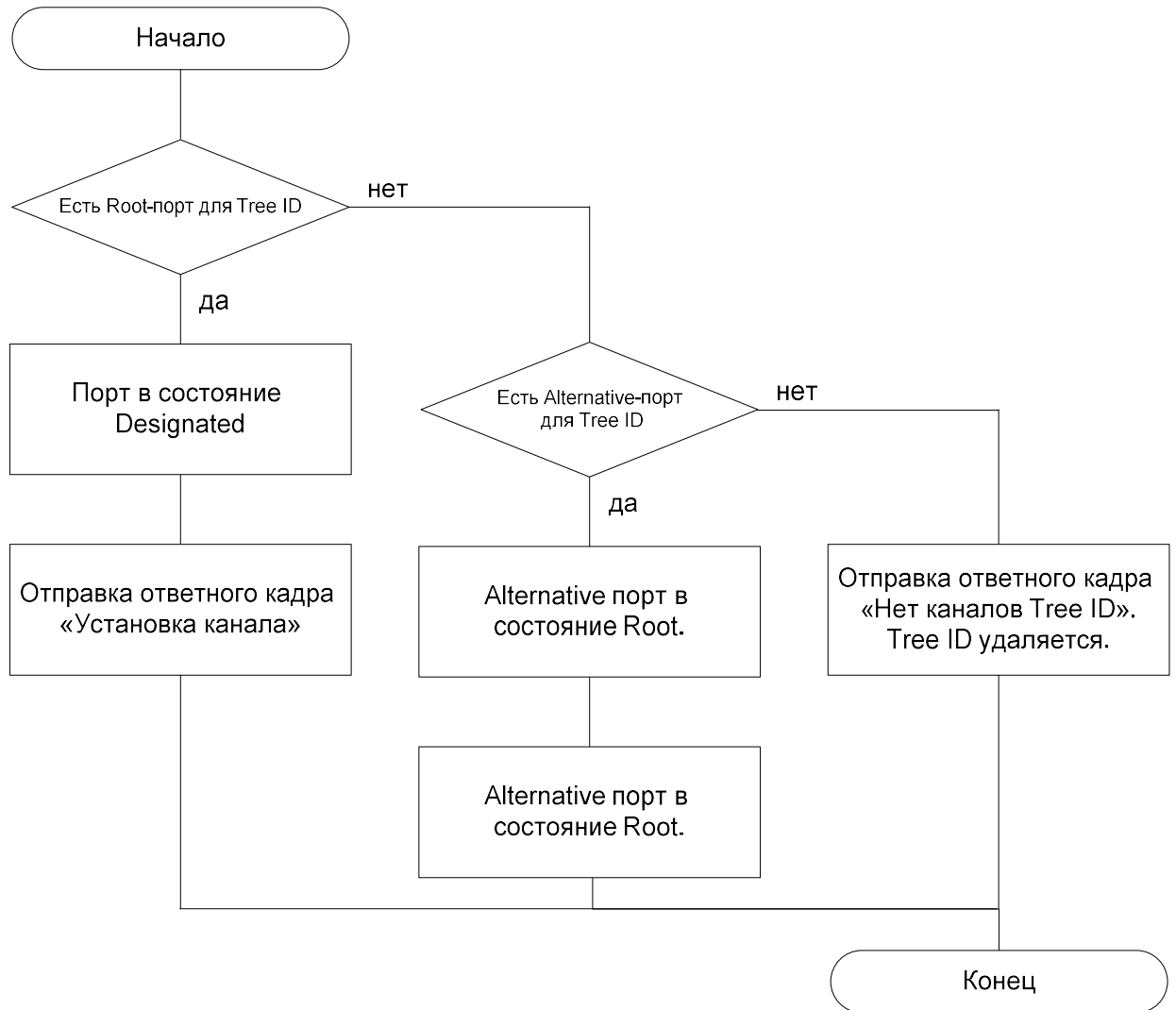
ПРИЛОЖЕНИЕ И

Обработка кадра «Предложение Root»



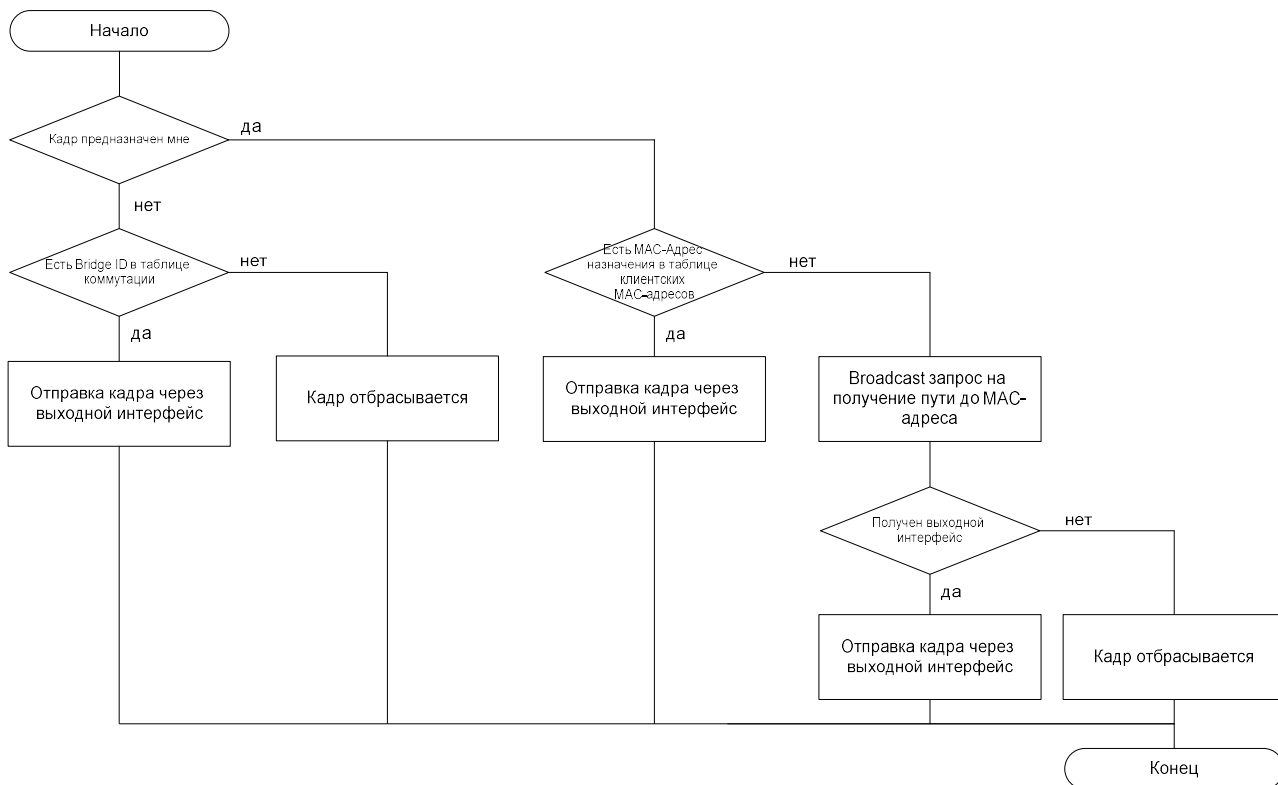
ПРИЛОЖЕНИЕ К

Обработка кадра «Потеряна Root-связь»



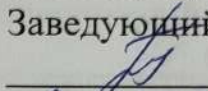
ПРИЛОЖЕНИЕ Л

Обработка кадра «Данные»



Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

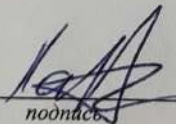
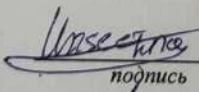
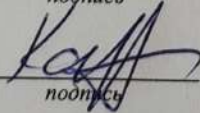
Институт космических и информационных технологий
Кафедра вычислительной техники

УТВЕРЖДАЮ
Заведующий кафедрой
 О.В. Непомнящий
«20» 06 2023 г.

БАКАЛАВРСКАЯ РАБОТА

090301 Информатика и вычислительная техника

Изучение сетевого протокола под языком C++ в Omnet++

Руководитель	 подпись	20.06.23 дата	доцент, канд. техн. наук должность, ученая степень	Ф.А. Казаков
Выпускник	 подпись	20.06.23 дата		И.И. Сошников
Нормоконтролёр	 подпись	20.06.23 дата	доцент, канд. техн. наук должность, ученая степень	Ф.А. Казаков

Красноярск 2023