

НАСТРОЙКА ТЕХНОЛОГИИ MST НА БАЗЕ ОТЕЧЕСТВЕННЫХ ОПЕРАЦИОННЫХ СИСТЕМ (ALT LINUX). ТЕСТИРОВАНИЕ.

В статье рассматривается практический аспект настройки технологии Multiple Spanning Tree (MST) для предотвращения петель на канальном уровне в среде отечественной операционной системы Alt Linux. Описан процесс развертывания лабораторного стенда на основе программного коммутатора, детально изложены этапы конфигурации MST-региона, экземпляров MSTI и привязки к ним VLAN. Экспериментальным путем подтверждена корректность работы настроенной конфигурации и возможность реализации отказоустойчивых сетевых топологий с использованием стандартных средств Linux без специализированного оборудования. Результаты работы демонстрируют применимость отечественных ОС для построения сложных сетевых инфраструктур.

1. ВВЕДЕНИЕ

При построении локальных вычислительных сетей с резервированием каналов связи особую актуальность приобретает задача предотвращения петель на канальном уровне модели OSI. Наличие петель приводит к возникновению широковещательных штормов, дублированию кадров и деградации работы сети в целом.

Для решения данной задачи используются протоколы семейства Spanning Tree, обеспечивающие построение логической древовидной структуры поверх физически избыточной топологии. Однако применение классического протокола STP не позволяет эффективно использовать сетевые ресурсы в сетях с виртуальными локальными сетями (VLAN), так как для всей сети формируется единое дерево.

Технология Multiple Spanning Tree (MST) является развитием протоколов Spanning Tree и позволяет формировать несколько независимых логических деревьев, к которым могут быть привязаны различные VLAN. Это

обеспечивает более рациональное использование каналов связи и повышает отказоустойчивость сети.

В данной работе рассматривается настройка технологии MST на базе отечественной операционной системы Alt Linux. В рамках исследования создаётся лабораторный стенд с программным коммутатором, выполняется настройка MST и проводится экспериментальная проверка корректности его работы.

Объект исследования — технологии предотвращения петель на канальном уровне.

Предмет исследования — настройка и функционирование Multiple Spanning Tree в среде Alt Linux.

2. ТЕОРЕТИЧЕСКАЯ ОСНОВА

Протоколы Spanning Tree предназначены для предотвращения образования петель в локальных сетях с избыточными соединениями. Основным принцип их работы заключается в логическом отключении избыточных каналов при сохранении физической резервируемости сети.

К основным разновидностям относятся:

STP (IEEE 802.1D) — классический протокол Spanning Tree. Формирует одно общее дерево для всей сети, имеет длительное время сходимости (до 50 секунд).

PVST+ (Per-VLAN Spanning Tree Plus) — проприетарное расширение Cisco для STP. Создает отдельный экземпляр дерева для каждого VLAN, но не является стандартом IEEE и создает высокую нагрузку на сеть ввиду большого количества служебных кадров.

RSTP (IEEE 802.1w) — ускоренная версия STP (Rapid STP). Значительно сокращает время реконвергенции сети (до 1-2 секунд), сохраняя обратную совместимость с STP, но, как и STP, строит единое дерево для всех VLAN.

MST (IEEE 802.1s) — развитие RSTP (Multiple Spanning Tree). Является стандартным решением, поддерживающим несколько логических деревьев (экземпляров MSTI), к которым можно привязать группы VLAN. Оптимален

для балансировки нагрузки и эффективного использования ресурсов в сетях с VLAN.

RSTP и MST используют схожие алгоритмы построения топологии и обмена служебными кадрами, однако MST обеспечивает более гибкое управление трафиком в сетях с VLAN.

Multiple Spanning Tree представляет собой метод организации нескольких экземпляров Spanning Tree в рамках одной сети. В MST VLAN объединяются в группы, каждая из которых обслуживается отдельным экземпляром дерева — MST Instance (MSTI).

Основные элементы MST:

MST Region — логическая область сети с едиными параметрами MST;

CIST (Common and Internal Spanning Tree) — общее дерево, обеспечивающее совместимость с другими STP-доменами;

MSTI — отдельные логические деревья внутри региона;

Привязка VLAN к MSTI — распределение VLAN между экземплярами MST.

Применение MST позволяет перераспределять нагрузку между каналами связи и повышать эффективность использования сетевой инфраструктуры.

Реализация протокола MSTP в дистрибутивах Linux, включая ALT, основана на разделении функций между ядром и пользовательским пространством. Подсистема bridge ядра Linux (конфигурируемая через утилиту ip link) отвечает за базовую коммутацию кадров, обработку тегов IEEE 802.1Q (VLAN) и предоставляет интерфейс для управления состояниями портов (Blocking, Learning, Forwarding) согласно указаниям управляющего демона.

Логика самого протокола MSTP (расчёт топологии, генерация и обработка BPDU, управление экземплярами MSTI) реализована в демоне пользовательского пространства. В рамках данного исследования использовался пакет mstpd версии 0.0.8, являющийся свободной реализацией

стандартов IEEE 802.1Q-2018 и 802.1s. Демон mstpd взаимодействует с подсистемой bridge через Netlink-сокеты, получая информацию о топологии и давая команды на изменение состояний портов моста. Для конфигурации используется утилита mstpcpl, позволяющая задавать параметры MST-региона, создавать экземпляры MSTI и выполнять привязку VLAN.

Таким образом, стек MSTP в ALT Linux представляет собой типичную для открытых систем модель: производительные механизмы обработки данных в ядре и гибкая логика управления в пространстве пользователя.

Официальная техническая документация, посвящённая стандартам IEEE 802.1, является ключевым источником информации о технологии MST. В ней описываются архитектура MST-регионов, принципы взаимодействия CIST и MSTI, а также правила привязки VLAN к экземплярам MST. Эти материалы используются для корректного понимания стандартизированных механизмов MST и их реализации в сетевых устройствах.

STP и RSTP работают на уровне физического канала, предотвращая мостовые петли при наличии избыточных путей. Однако, когда локальная сеть виртуализируется с помощью транкинга VLAN, каждая физическая ссылка представляет собой несколько логических соединений. Блокировка физической ссылки блокирует все ее логические ссылки и заставляет весь трафик через оставшиеся физические ссылки в дереве охвата. Резервные ссылки вообще не могут быть использованы. Более того, без тщательного проектирования сети, казалось бы, избыточные каналы на физическом уровне могут использоваться для подключения различных VLAN, и блокировка любой из них может отключить одну или несколько VLAN, что приведет к плохим путям. Вместо этого MSTP обеспечивает потенциально лучшее использование альтернативных путей, позволяя использовать альтернативные деревья для различных VLAN или групп VLAN. [5]

Отдельный класс источников составляют материалы, посвящённые реализации сетевых технологий в операционных системах семейства Linux. В данных публикациях рассматриваются возможности использования

программных коммутаторов, подсистемы bridge и механизмов VLAN для построения L2-топологий. Эти работы демонстрируют, что современные операционные системы могут использоваться в качестве полноценных сетевых узлов, реализующих функции коммутаторов и поддерживающих протоколы Spanning Tree.

Документация и практические руководства по администрированию Linux-систем описывают методы настройки программной коммутации, обработки VLAN и взаимодействия с протоколами предотвращения петель. Эти источники позволяют связать теоретические положения стандартов IEEE с их практической реализацией в среде отечественных операционных систем.

Так как одному экземпляру связующего дерева может быть сопоставлено множество VLAN, комитет, разрабатывающий стандарт IEEE 802.1s предложил развить концепцию MST. MST используется для привязки конкретной VLAN к конкретному экземпляру связующего дерева. Регион MSTP состоит из одного или нескольких коммутаторов с одинаковым идентификатором MSID(MST Configuration Identification) и локальной сети (конкретный коммутатор в регионе MSTP является назначенным (designated) коммутатором локальной сети, на коммутаторах, закрепленных за локальной сетью, протокол STP не работает). [2]

Для защиты топологий соединения различных регионов и отдельных коммутаторов строится общее покрывающее дерево (CST - Common Spanning Tree). В качестве корневой коммутатора в CST выбирается коммутатор с наименьшим приоритетом, а при равных с наименьшим ID. Каждый регион MSTP представляется для CST как отдельный виртуальный коммутатор.[4]

Когда область MST (MST region) подключена к топологии PVST+, MST «имитирует» PVST+ с помощью функции, называемой PVST simulation mechanism. Эта функция позволяет области MST отправить PVST+ BPDU (по одному для каждой VLAN) на интерфейсы, подключенные к коммутаторам PVST+. Эти BPDU содержат одну и ту же информацию и рассылают один и

тот же корневой мост. Интерфейсы, которые подключаются к топологии PVST+, называются граничными (boundary) интерфейсами/портами.[7]

Наконец, весь смысл внедрения MSTP заключается в том, чтобы позволить создавать различные пути пересылки для различных групп трафика. Для этого VLAN сопоставляются с «Экземплярами» MSTP, которые пронумерованы. По умолчанию все VLAN сопоставлены с экземпляром MSTP ноль, и поэтому без настройки каких-либо дополнительных экземпляров MSTP у вас остается одно общее Spanning Tree. Для выполнения балансировки нагрузки необходимо настроить по крайней мере один экземпляр MSTP, а затем сопоставить одну или несколько VLAN с этим новым экземпляром MSTP. Все VLAN, использующие этот экземпляр, будут придерживаться любого пути пересылки, созданного этим экземпляром.[6]

Все эти устройства имеют следующие характеристики:

- Включен протокол охватывающего дерева.
- То же название региона.
- Та же конфигурация сопоставления VLAN-to-instance.
- Тот же уровень пересмотра MSTP.
- Физически связаны вместе. [3]

Основные преимущества MSTP:

1. Избыточность: MSTP позволяет создавать несколько независимых связующих деревьев, что обеспечивает избыточность и защиту от петель в сети.

2. Улучшенная производительность: MSTP позволяет более эффективно использовать доступные пропускные способности сети, так как разные VLAN могут применять разные пути.

3. Гибкость: MSTP позволяет администраторам настраивать различные параметры для каждой виртуальной локальной области и оптимизировать сеть под конкретные требования.[1]

Таким образом, анализ литературы показывает, что технология Multiple Spanning Tree является хорошо проработанным и стандартизированным

решением для предотвращения петель в сетях канального уровня, а её реализация возможна не только на специализированном сетевом оборудовании, но и на базе современных операционных систем, включая отечественные дистрибутивы Linux2.

3. МЕТОДЫ ИССЛЕДОВАНИЯ

Исследование носит экспериментально-аналитический характер и направлено на практическое изучение настройки и функционирования технологии Multiple Spanning Tree.

В ходе работы использовались:

1. Одна виртуальная машина, выполняющая роль программного коммутатора;
2. Три виртуальные машины, выполняющие роль конечных устройств.

Целью данного исследования является практическое изучение технологии Multiple Spanning Tree и экспериментальная проверка возможности её реализации на базе отечественной операционной системы Alt Linux с использованием программного коммутатора.

В рамках исследования предполагается:

1. Изучить принципы работы MST и его взаимодействие с VLAN;
2. Реализовать программный коммутатор с поддержкой MST;
3. Подтвердить работоспособность настроенной конфигурации экспериментальным путём.

Реализация исследования осуществляется путём построения лабораторного стенда, включающего конечные устройства и программный коммутатор на базе Alt Linux. В качестве коммутирующего узла используется программный мост, реализованный средствами операционной системы.

В ходе реализации выполняются следующие этапы:

1. Подготовка программной среды, включающая настройку сетевой подсистемы и приведение системы к единому способу управления сетевыми интерфейсами.

2. Создание программного коммутатора, обеспечивающего коммутацию кадров между оконечными устройствами на канальном уровне.

3. Настройка виртуальных локальных сетей (VLAN) для логической сегментации трафика и подготовки основы для применения Multiple Spanning Tree.

4. Настройка технологии MST, включающая задание параметров региона MST, создание экземпляров MST и привязку VLAN к соответствующим экземплярам.

5. Экспериментальная проверка, заключающаяся в анализе состояний портов программного коммутатора и проверке передачи трафика между оконечными устройствами.

Реализация исследования ориентирована на использование стандартных средств операционной системы Alt Linux и не требует применения специализированного сетевого оборудования.

4. ПРОВЕДЕНИЕ ЭКСПЕРИМЕНТА И ЕГО РЕЗУЛЬТАТЫ

В рамках исследования была реализована простейшая топология, состоящая из одного коммутирующего узла и трёх оконечных устройств, соединённых между собой через программный коммутатор.

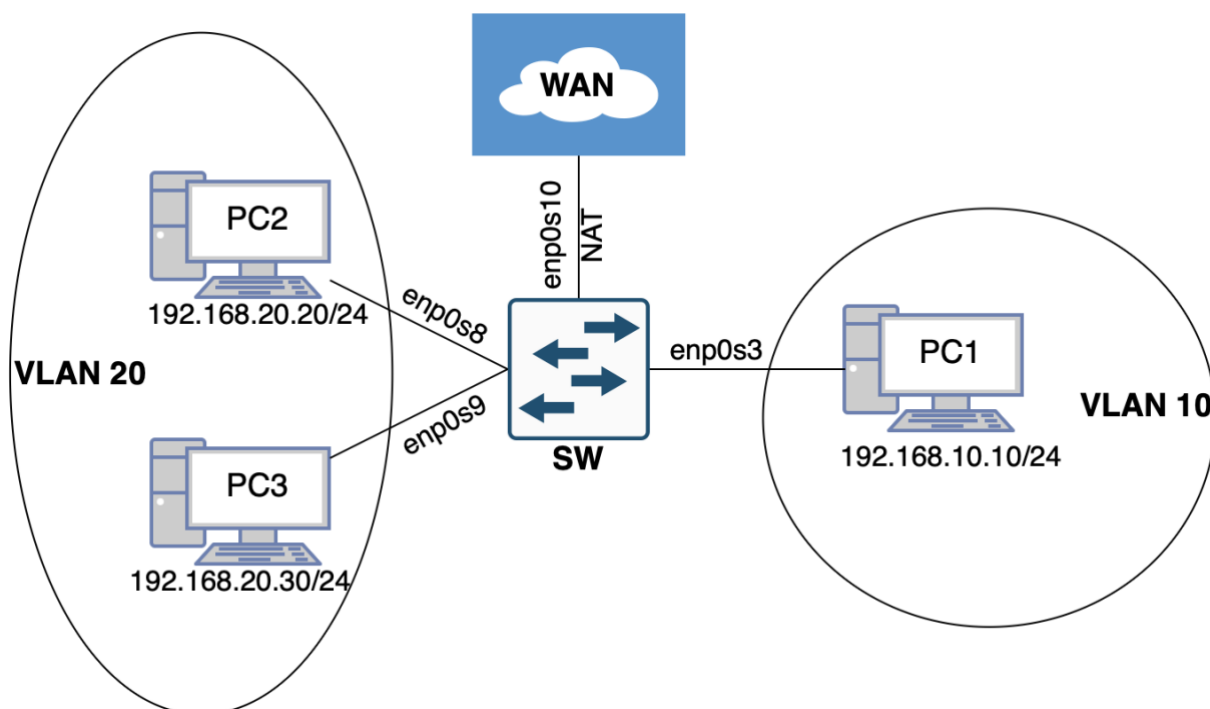


Рисунок 1 — Топология проведения эксперимента

В начале работы необходимо установить пакет mstpd и включить протокол, как показано на рисунках 2 и 3.

```
[root@sw ~]# apt-get update
Получено: 1 http://ftp.altlinux.org p11/branch/x86_64 release [4210B]
Получено: 2 http://ftp.altlinux.org p11/branch/x86_64-i586 release [16658]
Получено: 3 http://ftp.altlinux.org p11/branch/noarch release [2831B]
Получено 8706B за 0s (12,2kB/s).
Найдено http://ftp.altlinux.org p11/branch/x86_64/classic pkglist
Найдено http://ftp.altlinux.org p11/branch/x86_64/classic release
Найдено http://ftp.altlinux.org p11/branch/x86_64-i586/classic pkglist
Найдено http://ftp.altlinux.org p11/branch/x86_64-i586/classic release
Найдено http://ftp.altlinux.org p11/branch/noarch/classic pkglist
Найдено http://ftp.altlinux.org p11/branch/noarch/classic release
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
[root@sw ~]# apt-get install mstpd
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие НОВЫЕ пакеты будут установлены:
  mstpd
0 будет обновлено, 1 новых установлено, 0 пакетов будет удалено и 606 не будет о
бновлено.
Необходимо получить 83,9кВ архивов.
После распаковки потребуется дополнительно 223кВ дискового пространства.
Получено: 1 http://ftp.altlinux.org p11/branch/x86_64/classic mstpd 0.1.0-alt5:p
11+364191.100.1.1@1733441928 [83,9кВ]
```

Рисунок 2 — Установка пакета mstpd

```
[root@sw ~]# systemctl enable mstpd
Created symlink /etc/systemd/system/multi-user.target.wants/mstpd.service → /usr
/lib/systemd/system/mstpd.service.
[root@sw ~]# systemctl start mstpd
```

Рисунок 3 — Включение протокола mst

Помимо этого для работы протокола mst необходима первичная настройка моста и VLANов. Настройка IP адреса моста, например, как показано на рисунке 4.

```
[root@sw ~]# ip link set br0 type bridge stp_state 1
[root@sw ~]# ip link set dev br0 type bridge vlan_filtering 1
[root@sw ~]# ip link set enp0s3 master br0
[root@sw ~]# ip link set enp0s8 master br0
[root@sw ~]# ip link set enp0s9 master br0
[root@sw ~]# ip addr add 192.168.100.1/24 dev br0
[root@sw ~]# ip link set dev br0 up
[root@sw ~]# ip link set dev enp0s3 up
[root@sw ~]# ip link set dev enp0s8 up
[root@sw ~]# ip link set dev enp0s9 up
```

Рисунок 4 — Настройка моста br0

Затем идёт перезагрузка сети и распределение устройств в разные VLANы.

```
[root@sw ~]# systemctl restart network
[root@sw ~]# bridge vlan add dev enp0s3 vid 10 pvid untagged
[root@sw ~]# bridge vlan add dev enp0s8 vid 20 pvid untagged
[root@sw ~]# bridge vlan add dev enp0s9 vid 20 pvid untagged
[root@sw ~]#
```

Рисунок 5 — Настройка VLAN

После создания коммутатора и его включения последовательно идут команды для реализации технологии MST. В начале требуется добавить мост в MSTP. Устанавливается MST Configuration Identifier для нашего коммутатора (указывается имя коммутатора, уровень ревизии и имя региона). Далее нужно создать инстанцию MSTP (MSTID 1) на коммутаторе. Устанавливается отображение VID (VLAN ID) на FID (Forwarding Instance ID) для MSTP-деревьях с идентификатором 100 и 200. В конце устанавливается отображение FID на MSTID (MST Instance ID) для MSTP-деревьях с идентификатором 1 и 2. Также проверяется отображение VID на FID и FID на MSTID. Это все основные команды для настройки MSTP, они продемонстрированы на рисунке 6.

```
[root@sw ~]# mstpctl addbridge br0
[root@sw ~]# mstpctl setmstconfid br0 0 CommonRegion
[root@sw ~]# mstpctl createtree br0 1
[root@sw ~]# mstpctl createtree br0 2
[root@sw ~]# mstpctl setvid2fid br0 100:10 200:20
[root@sw ~]# mstpctl setfid2mstid br0 1:100 2:200
[root@sw ~]# mstpctl showvid2fid br0
br0 VID-to-FID allocation table:
  FID 0: 1-9,11-19,21-4094
  FID 100: 10
  FID 200: 20
[root@sw ~]# mstpctl showfid2mstid br0
br0 FID-to-MSTID allocation table:
  MSTID 0: 0-99,101-199,201-4095
  MSTID 1: 100
  MSTID 2: 200
[root@sw ~]#
```

Рисунок 6— Создание двух ветвей

Чтобы проверить состояние моста MST используется команда с рисунка 7.

```
[root@sw ~]# mstpcctl showbridge br0
br0 CIST info
  enabled          yes
  bridge id        8.000.08:00:27:15:15:E7
  designated root  8.000.08:00:27:15:15:E7
  regional root    8.000.08:00:27:15:15:E7
  root port        none
  path cost        0          internal path cost  0
  max age          20          bridge max age      20
  forward delay    15          bridge forward delay 15
  tx hold count    6          max hops             20
  hello time       2          ageing time          300
  force protocol version mstp
  time since topology change 2262
  topology change count 1
  topology change  no
  topology change port  enp0s8
  last topology change port  enp0s9
[root@sw ~]#
```

Рисунок 7 — Проверка настроек моста

Тестирование производилось с помощью передачи пакетов между ПК, настроенных средствами MSTP, находящихся в одном интерфейсе MSTI (в одной ветви).

Демонстрация проверки работоспособности MST через ping и wireshark для конечных устройств в VLAN 20 представлены на рисунках 8-10.

```
[root@pc2 ~]# ping 192.168.20.30
PING 192.168.20.30 (192.168.20.30) 56(84) bytes of data:
 64 bytes from 192.168.20.30: icmp_seq=1 ttl=64 time=9.437 ms
 64 bytes from 192.168.20.30: icmp_seq=2 ttl=64 time=8.762 ms
 64 bytes from 192.168.20.30: icmp_seq=3 ttl=64 time=7.246 ms
 64 bytes from 192.168.20.30: icmp_seq=4 ttl=64 time=7.893 ms
 64 bytes from 192.168.20.30: icmp_seq=5 ttl=64 time=8.415 ms
 64 bytes from 192.168.20.30: icmp_seq=6 ttl=64 time=9.128 ms
^C
--- 192.168.20.30 ping statistics ---
 6 packets transmitted, 6 received, 0% packet loss, time 5128ms
 rtt min/avg/max/mdev = 7.246/8.480/9.437/1.536 ms
```

Рисунок 8 — Связность устройств через ping

Time	Source	Destination	Protocol	Length	Info
1 0.000000000	PLSSystemtec_02:30:...	Spanning-tree-(ToF...	STP	151	MStI, ROOT = 32/08/0/08:00:27:15:15:E7 COST = 0 P0RT = 0x0002
2 1.999945773	PLSSystemtec_02:30:...	Spanning-tree-(ToF...	STP	151	MStI, ROOT = 32/08/0/08:00:27:15:15:E7 COST = 0 P0RT = 0x0002
3 4.00302291	PLSSystemtec_02:30:...	Spanning-tree-(ToF...	STP	151	MStI, ROOT = 32/08/0/08:00:27:15:15:E7 COST = 0 P0RT = 0x0002
4 0.002917819	PLSSystemtec_02:30:...	Spanning-tree-(ToF...	STP	151	MStI, ROOT = 32/08/0/08:00:27:15:15:E7 COST = 0 P0RT = 0x0002
5 0.000062125	PLSSystemtec_02:30:...	Spanning-tree-(ToF...	STP	151	MStI, ROOT = 32/08/0/08:00:27:15:15:E7 COST = 0 P0RT = 0x0002
6 9.999922459	PLSSystemtec_02:30:...	Spanning-tree-(ToF...	STP	151	MStI, ROOT = 32/08/0/08:00:27:15:15:E7 COST = 0 P0RT = 0x0002
7 12.00000023	PLSSystemtec_02:30:...	Spanning-tree-(ToF...	STP	151	MStI, ROOT = 32/08/0/08:00:27:15:15:E7 COST = 0 P0RT = 0x0002
8 14.000000076	PLSSystemtec_02:30:...	Spanning-tree-(ToF...	STP	151	MStI, ROOT = 32/08/0/08:00:27:15:15:E7 COST = 0 P0RT = 0x0002
9 16.001001924	PLSSystemtec_02:30:...	Spanning-tree-(ToF...	STP	151	MStI, ROOT = 32/08/0/08:00:27:15:15:E7 COST = 0 P0RT = 0x0002
10 18.000030000	PLSSystemtec_02:30:...	Spanning-tree-(ToF...	STP	151	MStI, ROOT = 32/08/0/08:00:27:15:15:E7 COST = 0 P0RT = 0x0002
11 20.000190087	PLSSystemtec_02:30:...	Spanning-tree-(ToF...	STP	151	MStI, ROOT = 32/08/0/08:00:27:15:15:E7 COST = 0 P0RT = 0x0002
12 22.000000095	PLSSystemtec_02:30:...	Spanning-tree-(ToF...	STP	151	MStI, ROOT = 32/08/0/08:00:27:15:15:E7 COST = 0 P0RT = 0x0002
13 24.000000027	PLSSystemtec_02:30:...	Spanning-tree-(ToF...	STP	151	MStI, ROOT = 32/08/0/08:00:27:15:15:E7 COST = 0 P0RT = 0x0002
14 26.000105022	PLSSystemtec_02:30:...	Spanning-tree-(ToF...	STP	151	MStI, ROOT = 32/08/0/08:00:27:15:15:E7 COST = 0 P0RT = 0x0002
15 28.001093011	PLSSystemtec_02:30:...	Spanning-tree-(ToF...	STP	151	MStI, ROOT = 32/08/0/08:00:27:15:15:E7 COST = 0 P0RT = 0x0002
16 30.000320222	PLSSystemtec_02:30:...	Spanning-tree-(ToF...	STP	151	MStI, ROOT = 32/08/0/08:00:27:15:15:E7 COST = 0 P0RT = 0x0002

Рисунок 9 — Захваченный сетевой трафик wireshark

```
Spanning Tree Protocol
Protocol Identifier: Spanning Tree Protocol (0x0000)
Protocol Version Identifier: Multiple Spanning Tree (3)
BPDU Type: Rapid/Multiple Spanning Tree (0x02)
BPDU flags: 0x7e, Agreement, Forwarding, Learning, Port Role: Designated
Root Identifier: 32768 / 0 / 08:00:27:1b:1b:e7
Root Path Cost: 0
Bridge Identifier: 32768 / 0 / 08:00:27:1b:1b:e7
Port Identifier: 0x8002
Message Age: 0
Max Age: 20
Hello Time: 2
Forward Delay: 15
Version 1 Length: 0
Version 3 Length: 9b
MST Extension
```

Рисунок 10 — Анализ пакета MST

Экспериментальный анализ BPDU-кадров в Wireshark подтвердил корректную работу реализации MSTP (IEEE 802.1s) в среде Alt Linux: захваченные кадры соответствуют стандарту (Protocol Version Identifier: 3, BPDU Type: 0x02), при этом поле «Version 3 Length» имеет значение 90, что указывает на активное использование расширения MSTP и передачу конфигурационных данных региона, тогда как нулевое значение поля «Version 1 Length» свидетельствует об отсутствии устаревшей информации STP (802.1D), обеспечивая обратную совместимость без избыточной нагрузки; полное совпадение идентификаторов Root Identifier и Bridge Identifier (32768.0.08:00:27:1b:1b:0e) при нулевой стоимости пути доказывает выполнение алгоритма выбора корневого моста, где программный коммутатор функционирует как корневой мост CIST; порт с идентификатором 0x8002 находится в состоянии Designated/Forwarding, что соответствует ожидаемому поведению в топологии с единственным коммутатором, а наличие MST Extension верифицирует передачу параметров региона и экземпляров MSTI, демонстрируя готовность системы к работе в избыточных топологиях с поддержкой множественных деревьев.

Тестирование подтвердило корректность работы настроенной конфигурации MSTP. Успешный обмен ICMP-пакетами (рисунок 8) между хостами в одном VLAN демонстрирует базовую связность. Анализ трафика в Wireshark (рисунки 9-10) зафиксировал наличие специальных кадров BPDU (Bridge Protocol Data Unit), генерируемых демоном mstpd. Эти кадры, отправляемые по протоколу MSTP (рисунок 10), служат прямым доказательством работы алгоритма связующего дерева. В условиях данного

стенда протокол закономерно переводит все порты в состояние Forwarding, так как физические петли отсутствуют.

Главным результатом является успешное создание двух независимых экземпляров MSTI (MSTID 1 и 2) и распределение между ними VLAN 100 и 200 (рисунок 6), что моделирует основу для балансировки нагрузки в реальной избыточной сети. Таким образом, эксперимент верифицирует возможность полной настройки и запуска стандартного MSTP средствами отечественной ОС ALT Linux.

5. ЗАКЛЮЧЕНИЕ

В рамках выполненной работы была изучена технология Multiple Spanning Tree и реализована её полная программная конфигурация на базе отечественной операционной системы Alt Linux. Проведённый эксперимент, несмотря на ограничения лабораторного стенда, показал, что использование стандартных средств Linux (демон mstpd, подсистема bridge) позволяет корректно развернуть и настроить все ключевые элементы MSTP — регион, экземпляры MSTI и привязку VLAN. Анализ сетевого трафика подтвердил генерацию и обработку служебных кадров протокола.

Полученные результаты подтверждают возможность применения Alt Linux в качестве эффективной платформы для изучения, отладки и построения основ отказоустойчивых сетей канального уровня с использованием стандартных протоколов. Цель работы достигнута.

6. СПИСОК ЛИТЕРАТУРЫ

1. Уймин А. Г. Компьютерные сети. L2-технологии. Практикум для СПО, Москва: Ай Пи Ар Медиа, 2024. – 191 с. – ISBN 978-5-4497-2539-4. – EDN AXDYGU.
2. Настройка протокола MST:
[\[https://www.manualsdir.ru/manuals/291457/qtech-qsw-3500.html?page=136\]](https://www.manualsdir.ru/manuals/291457/qtech-qsw-3500.html?page=136)

3. MST protocol and feature details:

[https://arubanetworking.hpe.com/techdocs/AOS-CX/10.16/HTML/12_bridging_8400/Content/Chp_stp/mst.htm]

4. MSTP: [<https://nag.wiki/display/DOC/01.+MSTP>]

5. Multiple Spanning Tree Protocol:

[https://en.wikipedia.org/wiki/Multiple_Spanning_Tree_Protocol]

6. Deep dive into MSTP: [<https://ine.com/blog/deep-dive-into-multiple-spanning-tree-protocol-mstp-configuration-and-troubleshooting>]

7. MST, PVST и их зависимость:

[<https://blog.sedicomm.com/2020/10/22/protokoly-mst-pvst-i-ih-sovmestimost/>]