## 1. Администратор информационной системы

Администратор ИС – специалист, объединяющий технические (системное проектирование), управленческие (координация персонала) и операционные (диспетчеризация) функции.

Главная обязанность администратора сети — обеспечить доступ к серверу и сетевым сервисам локальной сети, которая должна быть всегда доступна и (что очень важно!) прозрачна для пользователей. Обеспечить постоянную доступность сети можно только правильным планированием технического обслуживания, профилактикой и ремонтом.

## 2. Служба администрирования

Сети являются сложными аппаратно-программными системами, их используют сотни и даже тысячи пользователей. В такой ситуации крайне актуальным становится снижение издержек на содержание сети.

Исследования показали, что:

- затраты на добавление и перемещение оборудования и рабочих станций в расчете на одно рабочее место в год в среднем составляют 500 долл.;
- потери от простоев в расчете на одну сеть — 60 тыс. долл. в год.
- 95% времени простоев — это время, затраченное на нахождение повреждения (дефекта).
- Простой сети грозит огромными потерями: час простоя обходится некоторым компаниям в 0,5 млн долл.

**В процессе эксплуатации возникают проблемы, которые можно условно разделить на три большие группы:**

1. проблемы коммутации и связи;
2. аппаратного и программного обеспечения клиентской части;
3. работоспособности программных и аппаратных средств сети.

**Для решения этих проблем в рамках службы технической поддержки создаются соответствующие группы:**

- группа связи,
- инженерная
- административная группы.

**В связи со спецификой решаемых проблем общий контроль остается за административной группой.**

- В небольших сетях (до 20 рабочих мест) со всеми проблемами могут справиться один-два человека.

- Для больших систем (более 100 рабочих мест) этого уже недостаточно.

**Численность инженерной группы напрямую зависит от производителя аппаратного обеспечения клиентских рабочих мест:**

- Для обслуживания 200 компьютеров Brand-Name хватает двух-трех инженеров.
- Для такого же количества «желтых» систем это число придется удвоить.

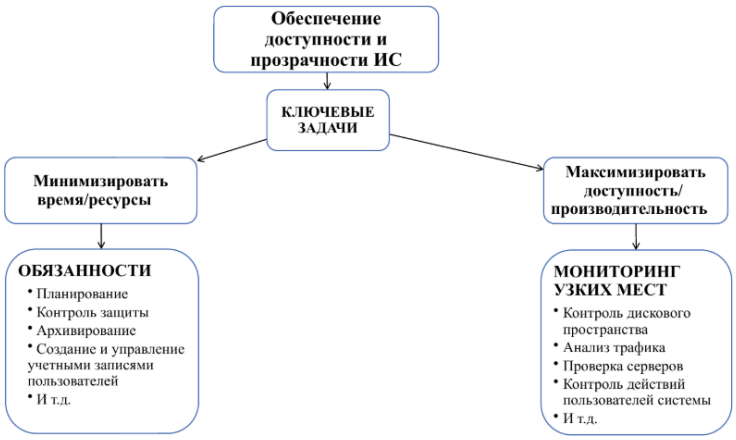
Для группы связи и административной группы такой подход неприменим. Их численность определяется на основе количества обслуживаемых устройств и регламента их повседневной деятельности.

В системе должен быть один администратор. На случай его отсутствия у одного из инженеров должен быть доступ в систему с администраторскими привилегиями. При большом количестве сетевых ресурсов численность административной группы может увеличиваться. На административную группу, как правило, возлагается и ряд дополнительных обязанностей (например, написание прикладного программного обеспечения).

### 3. Обязанности системного администратора

- Планирование системы и нагрузки.
- Установка/конфигурация аппаратных устройств и ПО.
- Контроль защиты.
- Архивирование (резервное копирование) информации.
- Создание и управление учетными записями пользователей.
- Определение и управление подсистемами.
- Управление системными ресурсами.
- Мониторинг производительности.
- Управление лицензиями.
- Документирование системной конфигурации и т.д.

**Как обязанности администратора обеспечивают главную цель**



## 4. Области сетевого администрирования

Сетевое администрирование распространяется на ряд основных областей, с которыми должен быть хорошо знаком администратор сети:

• управление пользователями — создание и поддержка учетных записей пользователей, управление доступом пользователей к ресурсам;

• управление ресурсами — установка, регистрация использования и поддержка сетевых ресурсов;

• управление конфигурацией — планирование конфигурации сети, ее расширение, а также ведение необходимой документации;

• управление производительностью и анализ эффективности — мониторинг и контроль за основными операциями для поддержания и улучшения производительности системами

• управление безопасностью — контроль доступа и сохранение целостности данных, процедуры аутентификации, проверка привилегий, поддержка ключей шифрования, управление полномочиями, управление паролями, внешним доступом, соединениями с другими сетями (именно из-за недостатков в этих механизмах происходит большинство вторжений в систему);

• поддержка — предупреждение, выявление и решение проблем сети.

## 5. Пирамида обязанностей сетевого администратора

В основании окажутся те обязанности, на исполнение которых уходит основная часть времени администратора. Это постоянные обязанности, осуществляемые в течение рабочего дня. Верхушка пирамиды — это операции, которые реализуются время от времени, но выполнение такой операции должно быть немедленным, а в некоторых случаях может занять день или более.

Пирамида обязанностей сетевого администратора слагается следующим образом.

1. Решение сетевых проблем.
2. Обучение пользователей.
3. Модернизация и замена компонентов сети.
4. Добавление в сеть новых компьютеров.
5. Модернизация программного обеспечения и установка нового.
6. Управление пользовательскими учетными записями.
7. Управление учетными записями групп.
8. Конфигурирование и обслуживание пользовательских настольных систем.
9. Анализ производительности, обеспечение защиты.

10. Обслуживание аппаратных и программных средств сервера и сети.

11. Планирование действий в аварийных ситуациях.

12 Резервное копирование и восстановление данных.

13. Планирование расширения сети.

Такая обязанность, как планирование расширения сети, предполагает анализ замечаний пользователей, прогнозирование и предупреждение их требований. Это постоянная и повседневная обязанность администратора.

Задача администратора — на основе собранной информации о производительности делать выводы, которые помогут принять решение о приобретении программ или оборудования.

Решение сетевых проблем («тушение пожаров») — срочная и требующая концентрации внимания обязанность с мгновенными последствиями. Например, при выходе из строя сетевой платы сервера необходимо действовать немедленно: отключить сервер, заменить плату, включить и протестировать его.

Если при конфигурации были учтены требования отказоустойчивости, возможно, в сети есть резервный сервер. В этом случае необходимо просто перевести пользователей на новый сервер, а затем заменить вышедший из строя.

## 6. Ежедневные задачи администрирования

1. Проверка файлов регистрации ошибок. Администратор должен просматривать новые записи во всех файлах регистрации ошибок. Эти записи могут поступать как из программы Event Manager, так и из программы Performance Monitor. Необходимо учитывать все предупреждения.

2. Проверка стола справок электронной почты. Администратор должен читать пользовательские запросы на получение справочной и иной информации, сортировать их по приоритетам и принимать ответные действия.

3. Проверка свободного пространства томов. Необходимо ежедневно отслеживать неоправданные «потери» и потенциальный дефицит дискового пространства. Этот процесс можно автоматизировать с помощью предупреждений (alerts) утилиты Performance Monitor.

4. Ежедневное создание резервных копий. Эго практикуется в связи с требованиями технологии работы предприятия. Если копии делаются ежедневно (это нужно делать либо ежедневно, либо ежедневно и еженедельно) и процедура резервирования автоматизирована (например, с помощью сервера Scheduler со встроенными средствами резервирования), необходимо следить за тем, чтобы в накопителе была установлена новая магнитная лента

соответствующего формата. Периодически нужно проверять и регулировать лентопротяжный механизм.

5. Проверка резервных копий, снимаемых ночью (если это делается). Необходимо проверять полноту и качество резервных копий, например, путем полной верификации или выборочной проверки файлов (в последнем случае нужно каждый раз выбирать разные файлы). Ленты желательно снимать и убирать в сейф. Не следует хранить единственный комплект резервных лент в одном . помещении с сервером — это чревато утратой всех данных (например, в случае пожара).

б. Контроль защиты. Основной особенностью любой сетевой системы является распределенность ее компонентов в пространстве, связь между которыми выполняется физически — с помощью сетевых соединений (коаксиальный кабель, витая пара, оптоволокно и т.д.) и программно — с помощью механизма сообщений. К сетевым системам наряду с обычными (локальными) атаками, осуществляемыми в пределах одной операционной системы, применим специфический вид атак, обусловленный распределенностью ресурсов и информации в пространстве, — так называемые сетевые (или удаленные) атаки. Они характеризуются тем, что, во-первых, злоумышленник может находиться за тысячи километров от атакуемого объекта, а во-вторых, нападению может подвергнуться не конкретный компьютер, а информация, передающаяся по сетевым соединениям.

## 7. Еженедельные задачи администрирования

- Удаление временных файлов: По возможности следует удалять из пользовательских и почтовых каталогов все неиспользуемые временные файлы.
- Создание и распространение списков использования дискового пространства пользователями: Необходимо проверять, как используется пространство на диске, и в случае необоснованного перерасхода напоминать ответственным пользователям, что им следует удалить или заархивировать свои файлы.
- Проверка состояния системы электронной почты: Нужно проверять, нет ли в системе чрезмерного количества уже прочитанных сообщений. Неиспользуемые резервные и другие временные файлы следует удалять.
- Создание резервных копий в конце недели (если это практикуется).

**8. Ежемесячные задачи администрирования.**

- Архивирование и удаление неиспользуемых файлов. Необходимо проверять, насколько интенсивно используются файлы. Если файлы не использовались более месяца, их нужно удалять или архивировать.
- Проверка готовности к устранению аварийных ситуаций. Эта проверка позволяет оценить разницу между способностью к восстановлению после отказа в периоды пиковой нагрузки и неспособностью вообще восстановить систему.
- Создание резервных копий в конце каждого месяца или в конце цикла (если это практикуется). Такая задача отличается от задачи ежедневного и еженедельного создания резервных копий. Ее суть — создание архива данных, циклических по своей природе. Например, подобную копию можно создавать в середине каждого второго месяца, если именно в это время заканчивается очередной технологический цикл. Архивируемые данные будут содержать итоговый отчет по каждому циклу. Хранить такие архивы нужно в другом помещении, чтобы ими можно было воспользоваться для восстановления данных после аварии.

**9. Эпизодические задачи администрирования.**

**1. Подключение и удаление аппаратных средств.**

Любая компьютерная сеть включает три основных компонента: активное оборудование (концентраторы, коммутаторы, сетевые адаптеры и др.), коммуникационные каналы (кабели, разъемы), сетевую операционную систему.

- Компоненты компьютерной сети: Активное оборудование, коммуникационные каналы, сетевая ОС.
- Ключевая задача: Конфигурация системы для распознавания/использования нового/подключенного оборудования (от принтера до диска).
- Оптимизация производительности системы (ключевые узлы - компьютеры): Включает две достаточно независимые задачи:
- Оптимизация ПК как элемента сети (тип сетевого адаптера, размер файлового кэша, производительность дисков и дискового контроллера, быстродействие центрального процессора и т.п).
- Оптимизация параметров протоколов, установленных на ПК (TTL IP, размер окна неподтвержденных пакетов, размер используемых кадров).

2. **Инсталляция новых программных средств.** После приобретения нового программного обеспечения его нужно инсталлировать и протестировать. Если программы

работают нормально, необходимо сообщить пользователям об их наличии и местонахождении.

Как правило, самой ответственной и самой сложной задачей системного администратора являются инсталляция и конфигурирование операционной системы. Во многих современных операционных системах разработчики идут по пути исключения многих непродуктивных параметров системы, с помощью которых администраторы способны влиять на производительность ОС. Вместо этого в операционную систему встраиваются адаптивные алгоритмы, которые определяют рациональные параметры системы во время ее работы. С помощью этих алгоритмов ОС может динамически оптимизировать свои параметры в отношении многих известных сетевых проблем, автоматически перераспределяя свои ресурсы и не привлекая к решению администратора.

Обычно при оптимизации производительности ОС администратор начинает этот процесс при заданном наборе ресурсов. В общем случае одновременно улучшить все критерии производительности невозможно. Например, если целью является увеличение доступной оперативной памяти, то администратор может увеличить размер страничного файла, но это приведет к уменьшению доступного дискового пространства.

После инсталляции и оптимальной настройки операционной системы начинается, практически бесконечный процесс установки программного обеспечения. И здесь на первый план выходят проблемы совместимости различных программ и безопасности.

3. **Подключение и удаление пользователей, Оказание им помощи**. Создание бюджетов для новых пользователей и удаление бюджетов тех пользователей, которые уже не работают, — обязанности системного администратора. Процесс включения и удаления пользователей можно автоматизировать, но некоторые решения, от которых зависит включение нового пользователя, должен принимать администратор.

Очень часто сотрудники предприятия оказываются самым слабым звеном в системе его безопасности, поэтому системному администратору следует уделять больше внимания работе с пользователями системы. Иначе простой листочек бумаги с паролем, лежащий на рабочем месте забывчивой сотрудницы, сделает бесполезной выверенную настройку вашего межсетевого экрана.

Для усиления безопасности компьютерных систем компании разумными могут считаться следующие шаги:

- привлечение внимания людей к вопросам безопасности;
- осознание сотрудниками всей серьезности проблемы и принятие в организации политики безопасности;

- изучение и внедрение необходимых методов и действий для повышения защиты информационного обеспечения.

В крупной (более 100 человек) организации для определения уровня ее защищенности можно провести тест на проникновение. Этот метод позволяет выявить недостатки безопасности с точки зрения постороннего человека. Он позволяет протестировать схему действий, которая раскрывает и предотвращает внутренние и внешние попытки проникновения и сообщает о них.

Тест должен разрешить два основных вопроса:

- Все ли пункты политики безопасности достигают своих, целей и используются так, как было задумано?
- Существует ли что-либо, не отраженное в политике безопасности, что может быть использовано для достижения злоумышленником своих целей?

4. **Поиск неисправностей.** Операционные системы и аппаратные средства, на которых они работают, время от времени выходят из строя. Задача администратора — диагностировать сбои в: системе и в случае необходимости вызвать специалистов. Как правило, найти неисправность бывает намного сложнее, чем устранить ее.

## 10. Процессы поддержки ИТ-сервисов

(Ниже расписаны) Блок процессов поддержки ИТ-сервисов включает следующие процессы:

1. Управление инцидентами
2. Управление проблемами
3. Управление конфигурациями
4. Управление изменениями
5. Управление релизами

## 11. Управление инцидентами

Процесс управления инцидентами предназначен для обеспечения быстрого восстановления ИТ-сервиса. При этом инцидентом считается любое событие, не являющееся частью нормального функционирования ИТ-сервиса. К инцидентам относятся, например, невозможность загрузить операционную систему, сбой электропитания, сбой жесткого диска на рабочей станции пользователя, появление компьютерного вируса в локальной сети офиса, отсутствие тонера или бумаги для печатающего устройства и т.д. **Показателями качества реализации процесса являются:**

- временная продолжительность инцидентов;

- число зарегистрированных инцидентов.

**При реализации процесса должны выполняться следующие**

- прием запросов
- регистрация
- категоризация
- приоритизация
- изоляция
- эскалация
- отслеживание развития
- разрешение
- уведомление
- закрытие инцидентов.

Необходимым элементом обеспечения эффективного функционирования процесса является создание службы поддержки пользователей (Help Desk), единой точки обращения по поводу различных ситуаций в ИТ-инфраструктуре, обработки и разрешении пользовательских запросов. Следует отметить, что роль службы поддержки пользователей в последнее время возрастает, что отражается в её модифицированном названии – Service Desk. Это говорит о том, что современные службы поддержки переориентируются с реактивного принципа работы, на проактивный, позволяющий анализировать ситуацию и предотвращать инциденты еще до их возникновения.

Для управления качеством процесса необходимо определить систему управления инцидентами, разработать управленческие отчеты и обеспечивать непрерывное улучшение процесса.

## 12. Управление проблемами

Процесс управления проблемами предназначен для минимизации негативного влияния инцидентов на бизнес и уменьшения количества инцидентов, за счет предотвращения возможных причин инцидентов. В данном контексте под проблемой понимают инцидент или группу инцидентов, имеющих общую неизвестную причину.

**При реализации процесса должны выполняться следующие**

- анализ тенденций
- регистрация
- идентификация корневых причин

- отслеживание изменений
- выявление известных
- управление известными
- решение
- закрытие проблем

Для управления качеством процесса необходима организация системы управления проблемами/известными ошибками, организация превентивных процедур поддержки, организация способов верификации известных ошибок, организация интерфейса поддержки поставщиком, разработка отчетов для управления, постоянное усовершенствование процесса.

### 13. Управление конфигурациями

**Цель:** Процесс управления конфигурациями предназначен для оказания помощи в управлении экономическими характеристиками ИТ-сервисов (комбинация требований клиентов, качества и затрат) за счет поддержания логической модели инфраструктуры ИТ и ИТ-сервисов, а также предоставление информации о них другим бизнес-процессам.

Процесс управления конфигурациями отвечает за поддержание информации о взаимоотношениях между конфигурационными единицами (CI) и за стандартизацию (CI), мониторинг информации о статусе (CI), их местоположении и всех изменениях (CI).

Информация о (CI) хранится в базе данных конфигурационных единиц (Configuration Management Data Base – CMDB).

CMDB — База данных управления конфигурациями представляет собой репозиторий метаданных, описывающий элементы конфигурации, их взаимосвязи и атрибуты.

**Функции процесса:**

- Идентификация и регистрация CI
- Мониторинг статуса и изменений CI
- Контроль целостности данных в CMDB
- Верификация соответствия CMDB реальному состоянию
- Предоставление информации другим процессам (например, управлению изменениями)

### 14. Управление изменениями.

Процесс управления изменениями предназначен для обеспечения уверенности ИТ-менеджера в том, что все изменения необходимы, запланированы и согласованы. Данный

процесс предполагает регистрацию всех существенных изменений в среде ИС предприятия, разрешает изменения, разрабатывает график работ по изменениям и организует взаимодействие ресурсов, всесторонне оценивает воздействие изменения на среду ИС и связанные с ним риски.

Основная задача данного процесса - проведение только обоснованных изменений в ИТ-инфраструктуре и отсев непродуманных или потенциально рискованных изменений. Для этого каждое изменение конфигурации ИС организации в обязательном порядке оформляется запросом на изменение. Запрос на изменение проходит стандартную процедуру одобрения. В зависимости от масштаба изменения решение принимается на уровне менеджера процесса, комитета по оценке изменений в рамках службы ИС, правления организации.

Конечный результат процесса — набор изменений, согласованных между собой и с существующей конфигурацией информационной системы и не нарушающих функционирования уже существующих сервисов. Все изменения в обязательном порядке регистрируются процессом управления конфигурацией.

**Процесс управления изменениями выполняет следующие**

- обрабатывает запросы на
- оценивает последствия
- утверждает
- разрабатывает график проведения изменений, включая восстановление при
- устанавливает процедуру обработки запроса на
- устанавливает категории и приоритеты
- управляет проектами
- организует работу комитета по оценке
- осуществляет постоянное улучшение процесса.

## 15. Управление релизами.

**Цель:** Процесс управления релизами предназначен для обеспечения согласованности изменений, вносимых в ИТ-инфраструктуру предприятия.

Под релизом понимается набор новых и/или измененных позиций конфигурации, которые тестируются и внедряются совместно.

Процесс управления релизами предполагает консолидацию, структурирование и оптимизацию всех изменений или обновлений, а также снижение риска при переводе сервиса на новый качественный уровень.

**Процесс управления релизами состоит из трёх этапов:**

- Разработка (опционально);
- Тестирование (обязательно) - необходимо определить критерии, по которым будет проводиться тестирование для каждого релиза, что позволяет определить степень готовности релиза к распространению и внедрению.
- Распространение и внедрение.

**Процесс управления релизами выполняет следующие**

- планирование
- проектирование, разработка, тестирование и конфигурирование
- подписание релиза в
- подготовка релиза и обучение
- аудит оборудования и ПО до начала внедрения изменений и по завершении
- размещение эталонных копий ПО в
- установка нового или усовершенствованного оборудования и
- постоянное улучшение процесса.

**По масштабу релизы подразделяются на три вида:**

- Большой (новая функциональность) - обычно содержит значительный объем новой функциональности, которая делает ранее сделанные исправления проблем частично или полностью избыточными. Также большой релиз обычно отменяет предшествующие малые релизы;
- Малый (незначительные улучшения)
- Чрезвычайный (исправления некоторого числа известных ошибок)

**По способу реализации релизы подразделяются также на три вида:**

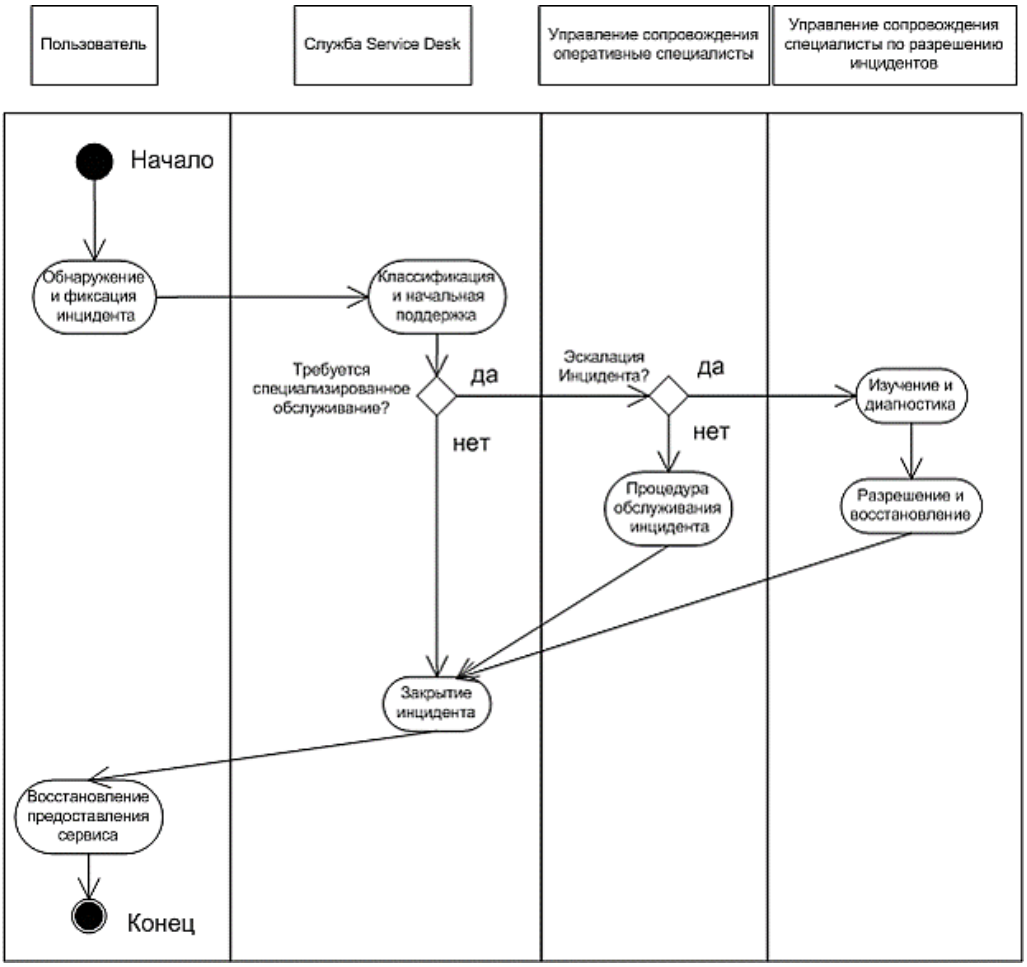
- Полный (все компоненты релиза разрабатываются, тестируются, распространяются и внедряются вместе)
- Дельта-релиз (включает в себя только новые или измененные позиции конфигурации)
- Пакетный (включает в себя несколько различных полных или частичных релизов, которые распространяются и внедряются совместно для снижения общего числа релизов, что облегчает работу пользователей)

## 16. Диаграмма активности процесса управления инцидентами

**Процесс управления инцидентами** предназначен для обеспечения быстрого восстановления ИТ-сервиса. При этом **инцидентом** считается любое событие, не

являющееся частью нормального функционирования ИТ-сервиса. К инцидентам относятся, например, невозможность загрузить операционную систему, сбой электропитания, сбой жесткого диска на рабочей станции пользователя, появление компьютерного вируса в локальной сети офиса, отсутствие тонера или бумаги для печатающего устройства и т.д. (ЭТОТ АБЗАЦ – ОПРЕДЕЛЕНИЯ ДЛЯ ПОНИМАНИЯ, ЧТО ЗА ПРОЦЕСС РАССМАТРИВАЕТСЯ НА ДИАГРАММЕ, МОЖЕТЕ ЕГО НЕ ПИСАТЬ)

Диаграмма активности процесса управления инцидентами




Пользователь ИТ-сервиса обнаруживает нарушение режима предоставления сервиса и обращается в Service Desk ИТ-службы. Сотрудник подразделения Service Desk фиксирует в регистрационном журнале инцидент, классифицирует его, определяет приоритет и при возможности осуществляет начальную поддержку. Если начальной поддержки пользователю достаточно и не требуется специализированная поддержка, то осуществляется закрытие инцидента. Если необходимо специализированное обслуживание, то информация по инциденту передается в подразделение сопровождения ИТ-сервисов. В этом подразделении на основе базы знаний выясняется возможность устранения инцидента оперативным персоналом, т.е. нет необходимости перемещения

инцидента на более высокий уровень обслуживания. В этом случае оперативный персонал реализует ранее документированную процедуру восстановления ИТ-сервиса.

Если для устранения инцидента отсутствует решение в базе знаний, то осуществляется перемещение на следующий уровень обслуживания, где специалисты высокого класса проводят изучение и диагностику инцидента, разрабатывают методы его устранения, восстановления заданной работоспособности ИТ-сервиса и пополняют базу знаний по инцидентам. После закрытия инцидента для пользователя предоставляется возможность доступа к ИТ-сервису с требуемыми показателями качества. Момент закрытия инцидента фиксируется в журнале службы ServiceDesk.

## 17. Функции процесса управления инцидентами

Процесс: Управление инцидентами

• Цель: Процесс управления инцидентами предназначен для обеспечения быстрого восстановления ИТ-сервиса.

• Инцидент: любое событие, не являющееся частью нормального функционирования ИТ-сервиса.

При реализации процесса должны выполняться следующие функции:

- прием запросов пользователей;
- регистрация инцидентов;
- категоризация инцидентов;
- приоритизация инцидентов;
- изоляция инцидентов;
- эскалация инцидентов;
- отслеживание развития инцидента;
- разрешение инцидентов;
- уведомление клиентов;
- закрытие инцидентов.

Необходимым элементом обеспечения эффективного функционирования процесса является создание службы поддержки пользователей (Help Desk), единой точки обращения по поводу различных ситуаций в ИТ-инфраструктуре, обработки и разрешении пользовательских запросов. Его основной принцип работы заключается в оперативном реагировании на обращения. Однако сейчас роль такой поддержки возрастает, что привело к развитию более широкой концепции Service Desk. Service Desk нацелен на бизнес и работает по проактивному принципу. Это означает, что он анализирует ситуацию и

предотвращает инциденты до их возникновения, обеспечивая поддержку на протяжении всего жизненного цикла услуги. Он использует целостный подход, согласованный с бизнес целями, и фокусируется на улучшении всего процесса поддержки, а не только на решении отдельных задач.

## 18. Функции процесса управления проблемами

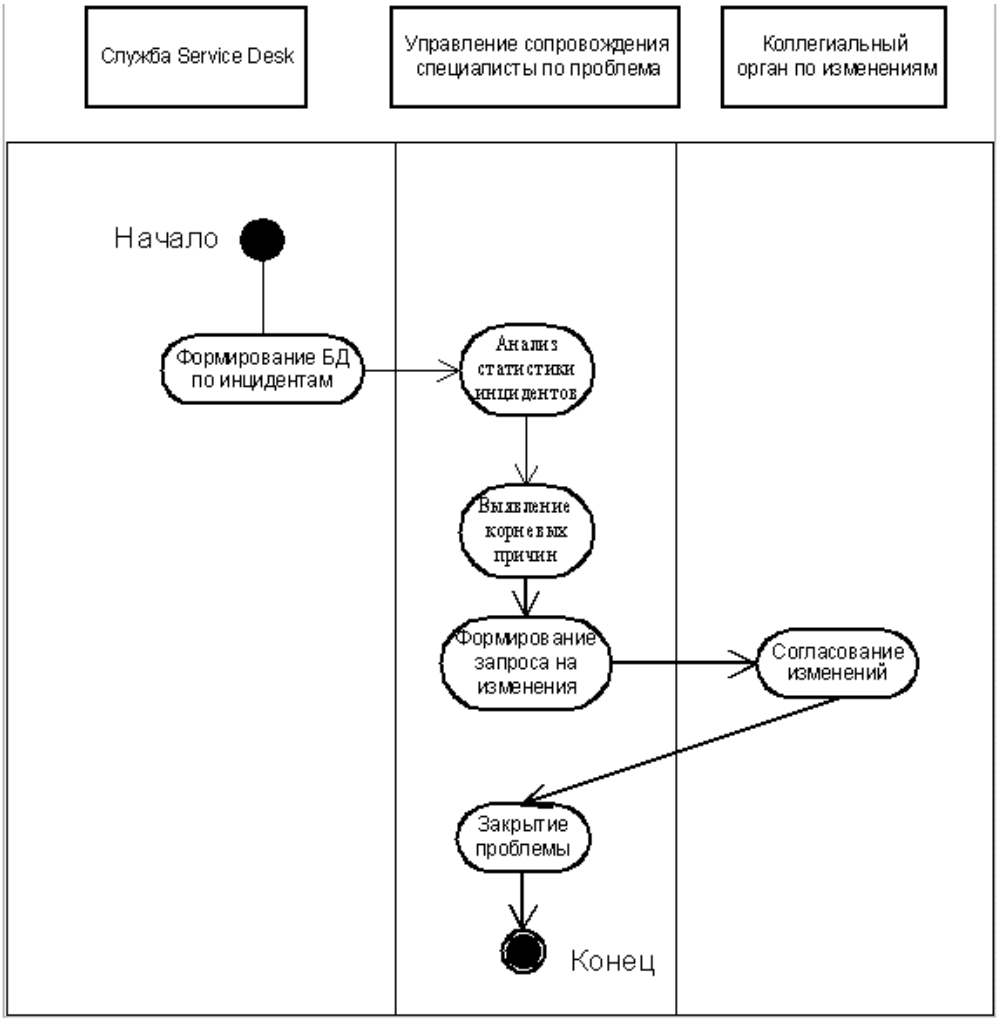
**Процесс управления проблемами** предназначен для минимизации негативного влияния инцидентов на бизнес и уменьшения количества инцидентов, за счет предотвращения возможных причин инцидентов. В данном контексте под **проблемой** понимают инцидент или группу инцидентов, имеющих общую неизвестную причину.

При реализации процесса должны выполняться следующие функции:

- анализ тенденций инцидентов;
- регистрация проблем;
- идентификация корневых причин инцидентов;
- отслеживание изменений проблем;
- выявление известных ошибок;
- управление известными ошибками;
- решение проблем;
- закрытие проблем.

Для управления качеством процесса необходима организация системы управления проблемами/известными ошибками, организация превентивных процедур поддержки, организация способов верификации известных ошибок, организация интерфейса поддержки поставщиком, разработка отчетов для управления, постоянное усовершенствование процесса.

**19. Диаграмма активности процесса управления проблемами**




На диаграмме изображен процесс управления проблемами, инициируемый службой Service Desk. Процесс начинается с формирования базы данных по инцидентам и последующего анализа их статистики для выявления корневых причин сбоев. После этого специалисты по проблемам и коллегиальный орган по изменениям формируют и согласовывают запрос на внесение необходимых изменений в ИТ-инфраструктуру. Конечной целью всего цикла является закрытие проблемы через внедрение корректирующих мер, что предотвращает её повторное возникновение.

**20. Функции процесса управления конфигурациями**

**Процесс управления конфигурациями** предназначен для управления экономическими характеристиками ИТ-сервисов за счет поддержания точной логической модели инфраструктуры и предоставления информации о ней другим бизнес-процессам. Это реализуется путём идентификации, мониторинга, контроля и обеспечения данных обо всех компонентах инфраструктуры – конфигурационных единицах (CI). Каждая такая

единица описывает конкретный элемент системы (например, сервер, программу или документ) с его атрибутами и версиями.

Процесс отвечает за поддержание информации о взаимосвязях между этими компонентами, их стандартизацию, отслеживание статуса, местоположения и изменений. Вся эта информация хранится в базе данных управления конфигурациями (CMDB), которая является центральным репозиторием метаданных, описывающих элементы, их связи и атрибуты для поддержки принятия управленческих решений.

При реализации процесса управления конфигурациями должны выполняться следующие функции:

- планирование – определение стратегии, правил и целей для реализации процесса, определение инструментария и ресурсов, определение интерфейсов с другими процессами, проектами, поставщиками;
- идентификация – разработка модели данных для записи в базу конфигураций всех компонент инфраструктуры ИТ, отношений между ними, а также информации о владельцах этих компонент, их статусе и соответствующей документации.

При спецификации процесса управления конфигурациями определяются его ключевые параметры:

- сфера охвата;
- глубина детализации;
- контроль;
- мониторинг статуса;
- верификация.

**Сфера охвата** (Scope) определяет, какая часть инфраструктуры будет находиться под контролем процесса. Например, можно охватывать только сервера и маршрутизаторы. Правильный выбор Сферы охвата очень важен на начальном этапе внедрения процесса Управление конфигурациями.

**Глубина детализации** (Level of Detail) – важный аспект, определяющий в дальнейшем отношения между CI. Отношения, как правило рассматриваются физические и логические.

Физические отношения:

- родители - дети;
- соединенная.

Логические отношения:

- копия;

- "использует", когда одна единица использует другую. Например, программа использует сервер.

**Контроль** процесса означает, что процесс контролирует все изменения, кем бы они не производились.

**Мониторинг статуса** предполагает отслеживание реального статуса CI, содержащихся в базе: в процессе жизненного цикла информационной системы статус CI может меняться от "заказано" до "исключено из конфигурации"

**Верификация** предполагает проверку того, насколько информация в базе конфигураций соответствует реальности.

При реализации процесса необходимо формировать отчеты руководству и другим процессам для осуществления их эффективного выполнения.

## 21. Классификация элементов конфигурации

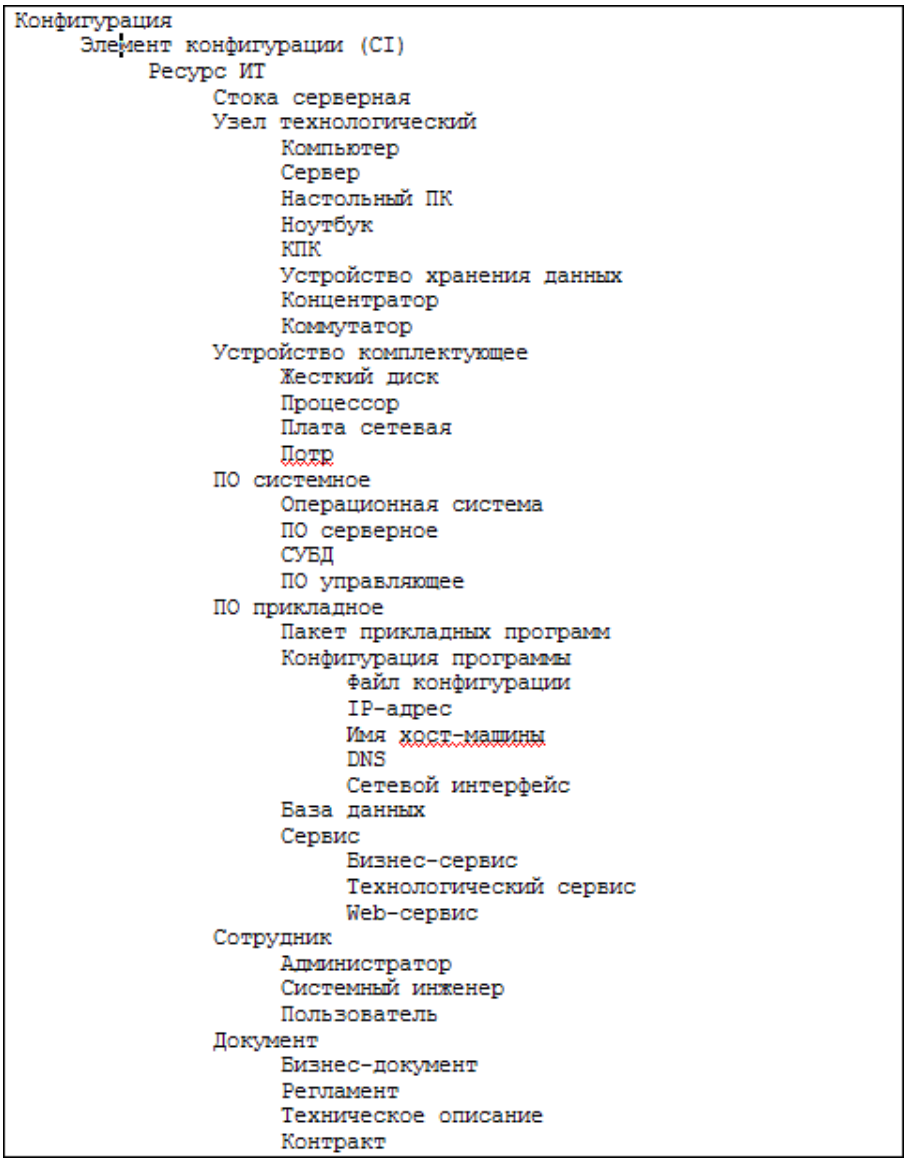
**Классификация элементов конфигурации** – это иерархическая структура, которая систематизирует все компоненты ИТ-инфраструктуры, управляемые в рамках процесса управления конфигурациями. Её цель – создать единую логическую модель для учёта, контроля и анализа взаимосвязей между элементами.

**Основные классы элементов конфигурации включают:**

- **Аппаратные ресурсы (ресурсы ИТ):** серверные стойки, технологические узлы, компьютеры (серверы, настольные ПК, ноутбуки), устройства хранения данных, сетевое оборудование (коммутаторы, концентраторы), а также комплектующие (жёсткие диски, процессоры, сетевые платы).
- **Программное обеспечение:** системное ПО (операционные системы), серверное ПО (СУБД), управляющее и прикладное ПО, включая пакеты прикладных программ, их конфигурации и конфигурационные файлы.
- **Сетевые и информационные элементы:** IP-адреса, доменные имена (DNS), сетевые интерфейсы и базы данных.
- **Сервисы:** бизнес-сервисы, технологические сервисы и веб-сервисы, которые являются конечным продуктом ИТ-подразделения.
- **Люди (персонал):** сотрудники, выполняющие определённые роли — администраторы, системные инженеры, пользователи.
- **Документация:** бизнес-документы, регламенты, технические описания и контракты.

Эта классификация служит основой для **Базы данных управления конфигурациями (CMDB)**, где каждый элемент описывается своими атрибутами (идентификатор, модель, статус, местоположение) и связями с другими элементами. Правильная классификация позволяет поддерживать целостность инфраструктуры, управлять изменениями, анализировать влияние сбоев и обеспечивать соответствие ИТ-услуг бизнес-требованиям.

ЕСЛИ БУДЕТ ВРЕМЯ И ЖЕЛАНИЕ, ПЕРЕРИСУЙТЕ ЭТУ КАРТТИНКУ В КАЧЕСТВЕ ПРИМЕРА




**22. Спецификации процесса управления конфигурациями**

Спецификации процесса управления конфигурациями определяют правила и требования, по которым осуществляется учет и контроль элементов конфигурации.

**К основным спецификациям относятся:**

– определение состава и границ конфигурации, то есть какие элементы включаются в управление конфигурациями;

– правила идентификации и наименования элементов конфигурации;

– требования к структуре и содержанию базы данных конфигураций;

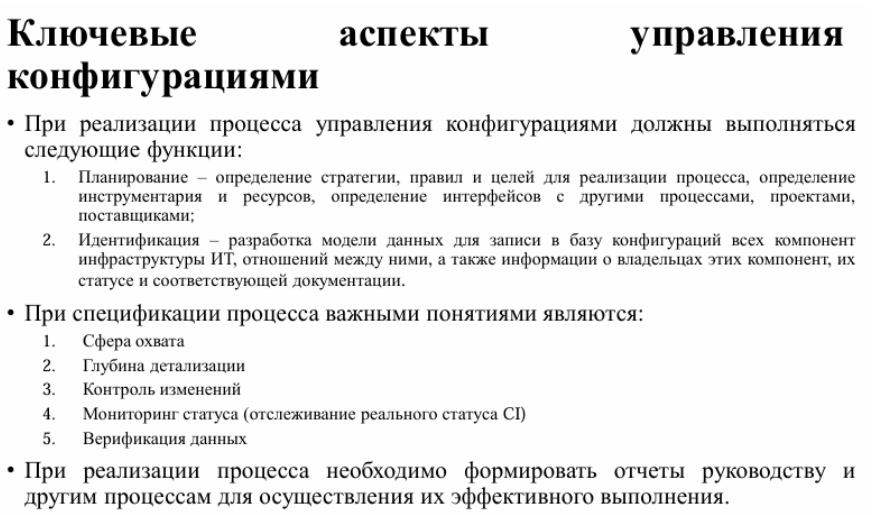
– порядок обновления информации при изменениях;

– требования к точности и актуальности данных;

– регламент проведения проверок и аудитов конфигураций;

– порядок взаимодействия с другими процессами управления ИТ-сервисами.

Четко определенные спецификации позволяют избежать хаоса в документации и обеспечивают воспроизводимость действий администраторов.




## 23. Функции процесса управления изменениями

Процесс управления изменениями предназначен для контроля всех изменений в ИТ-инфраструктуре с целью минимизации рисков и предотвращения сбоев.

**Цель управления изменениями** — обеспечить внесение изменений в ИТ-систему контролируемым, согласованным и предсказуемым образом.

**Основные функции процесса управления изменениями:**

- регистрация запросов на изменения;
- классификация изменений (стандартные, нормальные, аварийные);
- оценка влияния изменений на ИТ-сервисы и бизнес-процессы;
- анализ рисков и затрат, связанных с изменением;
- согласование изменений с ответственными лицами;
- планирование сроков и способов реализации изменений;
- контроль выполнения изменений;
- анализ результатов после внедрения изменения.

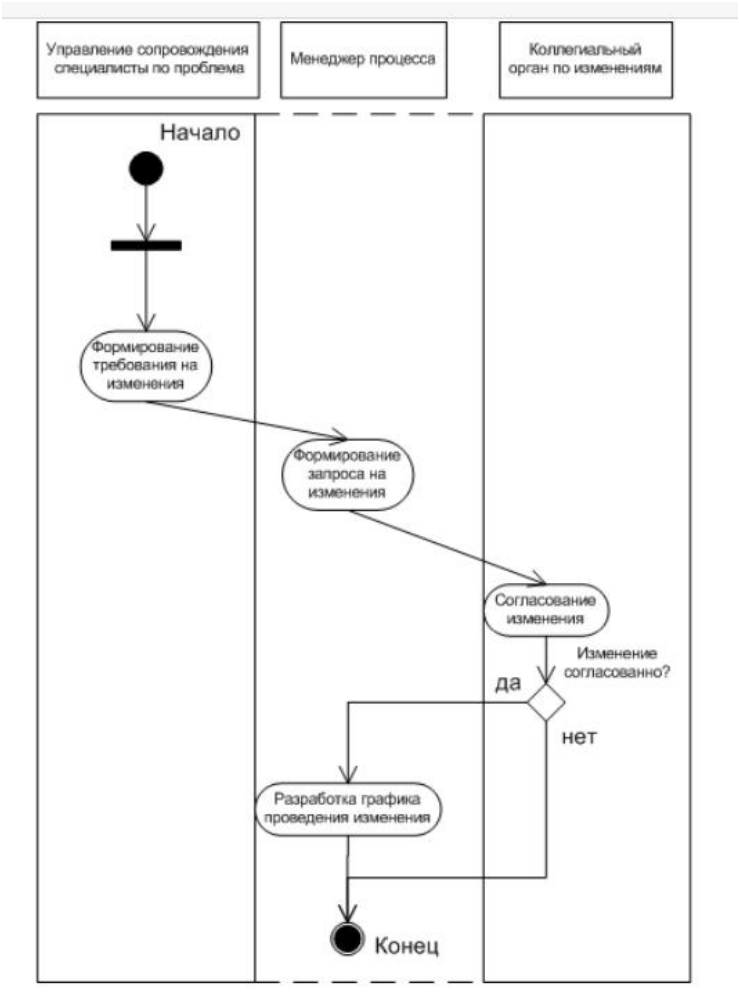
Управление изменениями является ключевым процессом для обеспечения стабильности системы, так как большинство крупных инцидентов возникает из-за неконтролируемых изменений.

### 24. Диаграмма активности процесса управления изменениям

Диаграмма активности процесса управления изменениями отражает последовательность действий от момента возникновения запроса на изменение до его завершения.

Процесс начинается с регистрации запроса на изменение. Далее запрос анализируется, классифицируется и оценивается с точки зрения рисков и влияния. После этого принимается решение о согласовании или отклонении изменения.

При одобрении изменения выполняется его планирование и реализация. После внедрения проводится проверка корректности работы системы и анализ результатов. Завершающим этапом является документирование изменения и обновление информации в системе управления конфигурациями.



**25. Функции процесса управления релизами**

Процесс управления релизами обеспечивает организованное внедрение изменений в рабочую среду.

**Цель управления релизами** — гарантировать, что новые или измененные компоненты ИТ-системы вводятся в эксплуатацию безопасно, стабильно и с минимальным воздействием на пользователей.

**Функции процесса управления релизами включают:**

- планирование релизов;
- формирование состава релиза;
- подготовку и тестирование релиза;
- документирование релиза;
- развертывание релиза в продуктивной среде;
- контроль корректности работы после внедрения;
- взаимодействие с пользователями и службами поддержки.

**26. Этапы процесса управления релизами**

Процесс управления релизами обеспечивает организованное внедрение изменений в рабочую среду.

**Цель управления релизами** — гарантировать, что новые или измененные компоненты ИТ-системы вводятся в эксплуатацию безопасно, стабильно и с минимальным воздействием на пользователей.

**Процесс управления релизами состоит из следующих этапов:**

- планирование релиза;
- разработка и сборка релиза;
- тестирование релиза;
- подготовка к внедрению;
- развертывание релиза;
- проверка и подтверждение успешного внедрения;
- закрытие релиза и анализ результатов.

Каждый этап направлен на снижение рисков и обеспечение стабильности системы.

**27. Способы реализации релизов**

Процесс управления релизами обеспечивает организованное внедрение изменений в рабочую среду.

**Цель управления релизами** — гарантировать, что новые или измененные компоненты ИТ-системы вводятся в эксплуатацию безопасно, стабильно и с минимальным воздействием на пользователей.

**Существует несколько способов реализации релизов:**

- поэтапное внедрение, при котором релиз разворачивается постепенно;
- параллельное внедрение, когда старая и новая версии работают одновременно;
- прямое внедрение, при котором новая версия полностью заменяет старую;
- пилотное внедрение, когда релиз сначала устанавливается на ограниченное число систем.

Выбор способа зависит от критичности системы и допустимых рисков.

### 28. Инсталляция FreeBSD, программа BSDInstall

**Инсталляция FreeBSD** — это процесс установки операционной системы FreeBSD на компьютер с последующей базовой настройкой системы для дальнейшей эксплуатации. Установка выполняется с использованием стандартной программы установки BSDInstall, которая входит в состав дистрибутива FreeBSD.

BSDInstall представляет собой текстовый интерактивный установщик, работающий в диалоговом режиме. Он позволяет пользователю пошагово выполнить все основные действия, необходимые для корректной установки операционной системы.

**Основные цели инсталляции FreeBSD:**

- установка базовой операционной системы;
- подготовка дисковой подсистемы;
- настройка сетевых параметров;
- создание пользователей и задание параметров безопасности;
- подготовка системы к дальнейшему администрированию.

**Процесс установки FreeBSD с помощью BSDInstall включает следующие основные этапы:**

1. Выбор языка и раскладки клавиатуры, что обеспечивает корректный ввод команд и текста.
2. Выбор типа установки (обычно стандартная установка).
3. Разметка диска, включая выбор схемы разделов и файловых систем. На этом этапе создаются системные разделы и точки монтирования.
4. Установка базовой системы FreeBSD, включающей ядро и основные

системные утилиты.

5. Настройка сети, включая выбор сетевого интерфейса, задание IP-адреса (автоматически или вручную) и параметров DNS.

6. Установка пароля суперпользователя root.

7. Создание учетных записей обычных пользователей.

8. Настройка системных служб и параметров безопасности.

9. Завершение установки и перезагрузка системы.

**29. Основные команды FreeBSD**

**ls**

Дать перечень файлов, находящихся в текущем каталоге. Эквивалент команды DIR, имеющейся во многих операционных системах. Команда ls -l дает более подробную информацию, включая размер файлов, их принадлежность и дату создания

**rm** *файлы*

DELETE. Стереть (удалить) один или несколько файлов. Например, команда rm file1 file2 file3 удаляет три файла: filel, file2, file3. Команда rm -i перед удалением каждого файла просит Вас подтвердить свое намерение.

**mv** *старое-имя новое-имя*

RENAME. Переименовать (переместить) файл из *старое-имя* в *новое-имя*.

**cp** *файл1 файл2*

СОРУ. Копировать *файл1* в новый файл с именем *файл2*.

**pwd**

Показать текущий каталог.

**cd** *каталог*

Изменить текущий каталог.

**mkdir** *каталог*

Создать новый каталог без файлов с именем каталог.

**rmdir** *каталог*

Стереть (удалить) каталог с именем каталог. Этот каталог должен быть пуст, т.е. в нем не должно быть файлов.

**man** *команда*

Вывести на экран справку ОС UNIX по команде команда.

**chmod** – изменение прав доступа

| Чьи права | Описание | Изменение | Описание | Доступ | Описание |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| a | Все триплеты | = | Присваивание | r | Чтение |
| u | Владелец | + | Добавление | w | Запись |
| g | Группа | - | Удаление | x | Выполнение |
| o | Остальные | | | | |

**chmod** –R g+w /usr/home/A

**30. Установка программного обеспечения с использованием пакетов и портов**

Установка приложений: порты и пакеты рассказывает о процессе установки программного обеспечения сторонних производителей с использованием "Коллекции Портов FreeBSD" и стандартных бинарных пакетов.

- Пакеты (pkg): Готовые скомпилированные бинарники.
- Плюсы: Быстро, не требует компиляции, не нужно знать процесс сборки.
- Минусы: Меньше вариантов настройки.
- Порты (/usr/ports/): Система для сборки из исходного кода.
- Плюсы: Гибкая настройка опций, актуальные версии, возможность применять патчи.
- Минусы: Долго, требует ресурсов, нужны исходники.

**Установка с использованием пакетов**

Основные команды:

- pkg update - Обновить список доступных пакетов
- pkg install <имя_пакета> - Установка пакета
- pkg remove <имя_пакета> - Удаление пакета
- pkg search <шаблон> - Поиск пакета
- pkg info - Список установленных пакетов
- pkg upgrade - Обновить все установленные пакеты

**Установка с использованием портов**

Основные команды:

- portsnap fetch update - Обновите дерево портов
- make install clean - Скомпилируйте и установите порт
- cd /usr/ports/... Перейти в каталог порта
- make install - Установить порт (без опций)
- make config - Настроить опции сборки
- make deinstall - Удалить установленный порт

make clean - Очистить временные файлы после сборки

**31. Стандарт иерархии файловой системы FHS**

**Стандарт иерархии файловой системы FHS (Filesystem Hierarchy Standard)**

- **История UNIX** — это путь множества конкурирующих ветвей развития. Это привело к серьезной проблеме: файлы в разных реализациях системы хранились в разных местах. Опыт работы с одной версией UNIX не переносился на другую, что создавало большие трудности для пользователей и администраторов.
- Частичным решением этой проблемы стало появление в 90-х годах FHS (Filesystem Hierarchy Standard — стандарт иерархии файловой системы).
- **Стандарт иерархии файловой системы (FHS)** – это документ, описывающий структуру каталогов и их расположение в Unix-подобных операционных системах, включая Linux. FHS определяет назначение каждого каталога, чтобы обеспечить единообразие и предсказуемость файловой системы для разработчиков и пользователей.
- У разных операционных систем есть одна общая черта - их основная файловая система. Для FreeBSD это Fast File System (или FFS), которая произошла от Unix File System (сокращенно UFS).

**Роль стандарта иерархии файловых систем (FHS). Классификация файлов**

- Текущая версия стандарта иерархии файловых систем - 3.0. Он был анонсирован 3 июня 2015 года.
- FHS определяет структуру каталогов и расположение файлов в операционных системах Linux и других Unix-подобных системах.
- **Ключевые критерии классификации:**
- **Статические и динамические файлы** : Частота изменений (программы / данные, логи).
- **Совместно используемые и локальные файлы**: Возможность использования по сети (NFS) (общие / локальные).
- **Цель классификации:** Определить, какие данные следует хранить в разделах, доступных только для чтения, и отделить каталоги, допускающие экспорт через NFS, от тех, которые не могут экспортироваться.

**32. Файловые серверы. Типы файловых серверов**

**Файловый сервер** — это специальный сервер или программное обеспечение, предназначенное для централизованного хранения, управления и обмена файлами в компьютерной сети. Он осуществляет функцию хранилища данных и обеспечивает пользователям доступ к файлам.

Типы файловых серверов:

- Сервер доставки файлов: Обеспечивает **пересылку** файлов между клиентом и сервером (FTP).
- Сервер совместного доступа к файлам: Обеспечивает **прямой доступ** к удаленной файловой системе, как к локальной (NFS, Samba)

**Серверы доставки файлов и серверы совместного доступа к файлам**

**Серверы доставки файлов (на примере FTP)**

- Суть: Копирование файлов с одного компьютера на другой.
- Преимущество: Универсальность и межплатформенность.
- Недостаток: Ограниченное управление файлами на сервере.
- Проблема: Невозможность напрямую редактировать файлы без скачивания.
- Безопасность: Пароли передаются в открытом виде (небезопасно).

**Серверы совместного доступа (NFS, Samba)**

- Суть: Файлы на сервере выглядят и работают как локальные.
- Преимущество: Прямое редактирование и загрузка файлов в приложения.
- Интеграция: Лучшая поддержка атрибутов файлов (UNIX, DOS).
- Использование: Чаще применяются в однородных средах (Windows/Samba, UNIX/NFS).

**Сравнение и перекрытие функций**

- Общее: Оба типа серверов позволяют копировать файлы.
- Гибридные клиенты: Некоторые клиенты могут использовать FTP для совместного доступа, а Samba — для доставки (smbclient).

**Критерии выбора:**

- Доставка (FTP): Для простой загрузки/скачивания, межплатформенность.
- Совместный доступ (NFS/Samba): Для непосредственной работы с файлами, в рамках одной ОС.
- Безопасность: Для доставки использовать SSH, а не FTP.

**33. Конфигурирование FTP-сервера**

**Конфигурирование клиентской и серверной части службы FTP**

**Способ запуска:** Серверы могут вызываться из сценариев запуска системы или через суперсервер inetd.

Конфигурационный файл: /etc/inetd.conf

Действия для активации:

Найдите и раскомментируйте строку:

#ftp stream tcp nowait root /usr/libexec/ftpd ftpd –l (Удалите символ # в начале)

Перезапустите суперсервер командой: killall -SIGHUP inetd

**Особенности стандартной конфигурации:**

- Доступ по логину и паролю системных пользователей
- Пользователи попадают в свои домашние каталоги
- Сохраняются стандартные права доступа Unix

**Важно:** Убедитесь, что порт 21/tcp не заблокирован фаерволом

**34. Конфигурирование FTP-клиента**

Стандартный клиент ftp уже предустановлен. Для подключения к серверу используйте команду в терминале:

ftp имя_сервера

Основные команды клиента:

- ls — список файлов на сервере
- cd — смена каталога на сервере
- lcd — смена локального каталога
- get файл — скачать файл
- put файл — загрузить файл
- binary — режим бинарной передачи (для программ, архивов)
- ascii — текстовый режим
- quit — выход

**35. Конфигурирование NFS-сервера**

Главный конфигурационный файл: /etc/exports

**Формат записи**: каталог [параметры] список_клиентов

**Ключевые параметры экспорта:**

- -alldirs: Позволяет клиентам монтировать любой подкаталог указанного каталога.
- -maproot=пользователь[:группа1[:группа2[:...]]]: Отображает права клиентского пользователя root на указанного пользователя сервера.
- -mapall=пользователь[:группа1[:группа2[:...]]]: Отображает права всех клиентских пользователей на указанного пользователя сервера.
- -ro или -o: Предоставляет доступ только для чтения.

**Список клиентов:**

- Можно указывать по имени (nova.luna.edu), IP-адресу (172.17.2.251), сети (-network 172.17.2.0 -netmask 255.255.255.0) или NIS-группе.
- **Внимание:** Отсутствие спецификации клиента открывает доступ для всех! Это опасно.

**Пример конфигурации и запуск NFS сервера**

**Пример файла /etc/exports:**

- /usr/src -network 172.17.2 -netmask 255.255.255.0 nova nebula blackhole \ browndwarf.luna.edu
- /home -alldirs 172.17.4.8 nova nebula
- /home -maproot=jennie blackhole

**Запуск NFS-сервера:**

- Сервер NFS реализован в виде демона nfsd. Он обычно вызывается из сценариев запуска системы.

1. В файле /etc/rc.conf добавьте строку: nfs_server_enable="YES"
2. Примените настройки командой: /etc/netstart (или перезагрузите систему)

**36. Конфигурирование NFS-клиента**

Если необходимо, чтобы пользователи UNIX-систем могли осуществлять чтение и запись файлов на сервере непосредственно из приложений, сконфигурируйте сервер NFS. Эго подразумевает создание экспортируемых файловых систем, т.е. каталогов, допускающих совместное использование.

Для каждого экспортируемого каталога должно быть задано следующее:

- имя каталога
- параметры, определяющие порядок обработки запросов сервером
- имена или IP-адреса компьютеров, которым разрешен доступ к каталогу

Вся эта информация содержится в конфигурационном файле сервера NFS: /etc/exports. экспортируемому каталогу в этом файле соответствует одна строка (строк может быть несколько, если каталог экспортируется с разными параметрами разным клиентам). Формат записей таков:

**каталог [параметры] список_клиентов**

В первом поле указывается полный путь к каталогу, например, /home/usr/X11R6. Спецификация каталога не должна содержать выражений ‘.’ (текущий каталог) и ‘..’ (родительский каталог). Кроме того, ни один из компонентов спецификации не должен являться символической ссылкой.

Параметры могут быть указаны или отсутствовать.

Последний компонент записи, представляет собой список клиентов, которым разрешен доступ к экспортируемому каталогу.

Спецификации клиентов разделяются пробелами. Поддерживается несколько клиентских спецификаций:

• **Имена узлов**. Разрешается указывать имена компьютеров, например, nova.luna.edu (или просто nova, если DNS-подсистема сервера ищет узлы в домене luna.edu).

• **IP-адреса**. Вместо имени компьютера может стоять IP-адрес, например, 172.17.2.251. В этом случае система будет работать чуть быстрее, поскольку не выполняется поиск в DNS.

• **Сетевые адреса**. Чтобы охватить все компьютеры сети, воспользуйтесь опцией -network, указав укороченный адрес сети. Можно также задать маску сети с помощью опции -netmask. Например, спецификация -network 172.17.2 -netmask 255. 255. 255. 0 заставляет сервер принимать запросы от всех клиентов в адресном диапазоне 172.17.2.0/24.

• **Сетевые группы NIS**. Если в сети функционирует сервер NIS, можно указать имя сетевой группы NIS. Вообще говоря, система всегда сначала пытается интерпретировать имя узла как имя сетевой группы.

• **Отсутствие спецификации**. Если клиентская спецификация отсутствует, сервер NFS принимает запросы от любых клиентов.

### 37. Почтовые серверы

Комьютер может функционировать в качестве почтового сервера. На одном компьютере может быть запущено несколько серверов, отвечающих за обработку различных почтовых протоколов.

Некоторые из этих протолов (известны как **протоколы принудительной доставки**) требуют, чтобы передачу почты инициировал отправитель.

В других протоколах (их называют **протоколами доставки по запросу**) начальный запрос поступает от получателя.

**Протоколы принудительной доставки:**

- Передачу инициирует отправитель
- Основной протокол: SMTP
- Получатель должен быть постоянно доступен
- Если доставка невозможна — сообщение временно хранится, затем удаляется

**Протоколы доставки по запросу:**

- Передачу инициирует получатель

- Основные протоколы: POP, IMAP
- Используются для рабочих станций, которые подключаются к Internet нерегулярно
- Обычно применяются на последнем этапе доставки

**Ретранслятор почты** — система, принимающая почту от одного компьютера и пересылающая на другой. Сообщение может проходить через несколько ретрансляторов.

**Типичная цепочка доставки:**

1. Пользователь создает письмо
2. Почтовый агент передает письмо серверу исходящей почты (SMTP)
3. Сервер через SMTP пересылает письмо другим серверам
4. Сервер домена получателя определяет адрес через DNS
5. Используется специальная DNS-запись MX — указывает сервер для приема почты
6. Последний сервер домена хранит почту
7. Получатель забирает письмо через POP/IMAP

**Назначение MX-записей**

MX (Mail Exchange) — DNS-записи, указывающие сервер, отвечающий за прием почты домена. Сервер-отправитель ищет MX для домена получателя и передает сообщение указанному серверу.

**Где применяется доставка по запросу.** Чаще всего — только на последнем этапе**.** Удобно при непостоянном подключении к Internet**.** Может применяться и в середине цепочки (например, при NAT или локальных сетях)

**Дополнительные серверы используются для:**

- разделения входящей и исходящей почты
- распределения нагрузки
- преобразования форматов почты

**Серверы принудительной доставки (SMTP)**

| Сервер | Особенности |
|---|---|
| Sendmail | Самый распространённый, стандартный для FreeBSD |
| qmail | Модульный, эффективный, безопасный, использует другой формат почты |
| Exim | Совместим с sendmail, мощная фильтрация, удобен против спама |
| Postfix | Модульный, безопасный, формат как у sendmail |

**38. Конфигурационные файлы sendmail**

**Основные конфигурационные файлы:**

- /etc/mail/sendmail.cf – основной конфигурационный файл sendmail
- /etc/mail/freebsd.mc – файл макросов, который редактируют администраторы

Файл sendmail.cf имеет очень сложный формат. Не рекомендуется редактировать напрямую, так как ошибка в нем может вывести сервер из строя. Редактируют файл freebsd.mc, затем на его основе создают sendmail.cf.

После редактирования freebsd.mc выполняется:

cd /usr/share/sendmail/cf/m4

m4 cf.m4 /etc/mail/freebsd.mc > /etc/mail/sendmail.cf

Затем sendmail перечитывает конфигурацию:

killall -SIGHUP sendmail

**Стандартная конфигурация sendmail по умолчанию:**

- принимает почту только для пользователей текущего компьютера
- отправляет почту непосредственно заданному компьютеру

Для почтового сервера домена эту конфигурацию нужно изменять.

По умолчанию sendmail принимает письмо только если имя компьютера в адресе совпадает с именем текущего компьютера.Чтобы сервер принимал почту для домена (например ben@threeroomco.com), используется файл:

/etc/mail/local-host-names

В стандартной установке отсутствует → его нужно создать. Он содержит список доменов, которые считаются локальными

Также необходимо:

корректно настроить DNS

чтобы MX-записи этих доменов указывали на данный сервер

Имя файла local-host-names задается в freebsd.mc через параметр confCW_FILE.

**39. Конфигурирование DNS-сервера**

DNS-сервер обслуживает две группы пользователей: внутреннюю и внешнюю.

**Внутренние пользователи**

- Компьютеры локальной сети

- Посылают запросы локальному DNS-серверу
- Сервер:
    - возвращает IP-адреса известных ему доменных имен
    - либо обращается к внешним DNS-серверам

**Внешние пользователи**

- Администраторы доменов
- Для обслуживания домена требуется минимум два DNS-сервера
- Часто используются внешние DNS-службы
- При наличии статических IP-адресов можно запускать собственные DNS-серверы
- В сложных конфигурациях DNS может интегрироваться с DHCP для динамического обновления записей

**Запуск собственного DNS-сервера оправдан, если:**

- в сети несколько десятков компьютеров
- компьютеры должны обращаться друг к другу по именам

Если один-два сервера и много клиентов, то DNS-сервер не требуется, достаточно файла /etc/hosts

**DNS-сервер во FreeBSD**

Наиболее распространённый сервер: BIND

Демон: named

Устанавливается во FreeBSD по умолчанию

Альтернативы: djbdns, dns_balance

Рассматривается именно BIND, из-за его распространенности

Описывается базовое конфигурирование для домашних и небольших сетей.

Ошибка конфигурации может привести к серьезным проблемам в работе домена.

**Запуск и управление сервером BIND**

- Включение автозапуска. В файл /etc/rc.conf: named_enable="YES"
- Запуск вручную: named
- Перечитывание конфигурации: killall -HUP named
- Полная перезагрузка с очисткой кэша: killall named; named

**40. Качество обслуживания в пакетных сетях. Типы QoS**

Качество обслуживания в пакетных сетях имеет статистический характер, потому что:

- пакеты поступают в сеть в случайные моменты времени,
- очереди и задержки — случайные процессы,
- скорость передачи и задержки пакетов — случайные величины.

Поэтому параметры QoS измеряются статистически:

- средняя скорость потока,
- средняя задержка,
- вариации (дисперсии) скорости и задержек,
- уровень потерь пакетов.

Оценки QoS получают путем усреднения параметров за заданный интервал времени.

**Типы QoS (по степени строгости гарантий):**

**1) Сервис по мере возможности (Best Effort)** (фактически — отсутствие QoS)

- Никаких гарантий
- Все пакеты обслуживаются одинаково
- Принцип обслуживания: FIFO
- Примеры: классические сети Ethernet, IP

**2) Сервис с предпочтением (мягкий QoS)**

- Некоторые типы трафика обслуживаются лучше, чем другие
- Обеспечивает:
- в среднем большую пропускную способность,
- меньшие задержки,
- меньшие потери для приоритетного трафика
- Нет численных гарантий
- Качество обслуживания зависит от текущей нагрузки сети
- Высокоприоритетный трафик может вытеснять низкоприоритетный

**3) Гарантированный сервис (жесткий QoS)**

- Предоставляет статистические численные гарантии
- Основан на предварительном резервировании ресурсов
- Гарантирует:
- пропускную способность,
- задержки,
- другие параметры QoS

Гарантии действуют при условии соблюдения источником оговоренных ограничений

Гарантии носят вероятностный характер (например, с вероятностью 0,999)

Обязательный элемент: входной контроль потоков

(чтобы источники не превышали согласованные параметры трафика).

**3. Совместное использование типов QoS**

Три типа QoS дополняют друг друга и могут использоваться совместно:

- Некритичный трафик → Best Effort
- Трафик с желаемым улучшением качества → Сервис с предпочтением
- Критичный трафик (например, видеоконференции, системы реального времени) → Гарантированный сервис

**41. Требования разных типов приложений к качеству обслуживания**

**Трафик приложений классифицируется по трём основным характеристикам:**

- Предсказуемость скорости передачи данных
- Чувствительность к задержкам пакетов
- Чувствительность к потерям и искажениям пакетов

**Требования по предсказуемости скорости:**

- Потоковый трафик (Stream, CBR)
- Равномерный поток данных
- Постоянная битовая скорость (CBR)
- Пакеты одинакового размера B с интервалом T
- Средняя скорость: CBR = B / T

**Пульсирующий трафик (Burst, VBR)**

- Непредсказуемая передача
- Периоды молчания сменяются пульсациями
- Переменная битовая скорость (VBR)
- Характеризуется коэффициентом пульсации
- У пульсирующих приложений коэффициент: 2:1 – 100:1
- У потоковых — существенно меньше

**Требования по чувствительности к задержкам**

- Асинхронные — задержки почти не важны. Пример: электронная почта
- Синхронные — чувствительны, но допускают задержки.

- Интерактивные — задержки заметны пользователю. Пример: удалённый текстовый редактор
- Изохронные — существует порог задержки. (для голоса: 100–150 мс)
- Сверхчувствительные — задержка разрушает функциональность. Пример: управление объектами в реальном времени.

**Грубое деление:**

- Асинхронные
- Синхронные (включая изохронные и сверхчувствительные)

**Требования по чувствительности к потерям пакетов**

- Чувствительные к потерям. Потеря даже малого фрагмента разрушает данные. Текст, программы, файлы, БД, электронная почта
- Устойчивые к потерям. Потери могут восстанавливаться по соседним данным. Большинство аудио- и видеоприложений. Допустимые потери — не более ~1%. Компрессированное видео и голос — чувствительны к потерям

**Типовые сочетания требований**

Характеристики могут сочетаться по-разному, но устойчивых сочетаний немного.

Пример распространённого класса приложений:

*Равномерный поток + изохронное + устойчивое к потерям*

→ *IP-телефония, видеоконференции, интернет-аудио*

Эта классификация лежит в основе требований QoS в современных сетях.

**42. Классификация приложений по чувствительности к задержкам пакетов**

**Основные классы (по возрастанию чувствительности):**

1. Асинхронные приложения. Практически нечувствительны к задержкам. Допускают очень большие задержки (вплоть до секунд). Пример: электронная почта

2. Синхронные приложения. Чувствительны к задержкам, но допускают их. Задержки влияют на работу приложения

3. Интерактивные приложения. Задержки заметны пользователю. Функциональность приложения не нарушается. Пример: текстовый редактор с удалённым файлом

4. Изохронные приложения. Существует порог задержки, при превышении которого резко ухудшается работа. Для передачи голоса критическая задержка: 100–150 мс

5. Сверхчувствительные к задержкам приложения. Задержка доставки данных разрушает функциональность. Пример: системы управления в реальном времени

**Упрощённая классификация. Все приложения делят на:**

- Асинхронные
- Синхронные

К синхронным в широком смысле относятся:

- изохронные,
- сверхчувствительные,
- часть интерактивных приложений.

**43. Параметры качества обслуживания**

В лекциях и презентациях есть два определения этих параметров. Первое, имеющее отношение к QoS в целом, взято из пятой лекции:

**Качество обслуживания (QoS) в пакетных сетях — статистическая характеристика.**

Все параметры, которыми измеряется качество обслуживания в пакетных сетях, являются статистическими.

- средние значения (математическое ожидание),
- вариации (дисперсии) скорости информационного потока,
- задержек пакетов.

Числовые оценки качества обслуживания могут быть на практике измерены путем усреднения соответствующих величин на каком-либо заранее оговоренном промежутке времени.

Второе, относящееся уже к критериям оценки приложений, взято уже из шестой лекции и, вероятно, подходит лучше под вопрос:

Трем критериям классификации приложений (предсказуемость скорости передачи данных, чувствительность к задержкам и чувствительность к потерям и искажениям) соответствуют три группы параметров требуемого качества обслуживания.

1. **Параметры пропускной способности**. Средняя, максимальная (пиковая) и минимальная скорости.
2. **Параметры задержек**. Средняя и максимальная величины задержек, среднее и максимальное значения вариаций задержек.
3. **Параметры надежности передачи.** Процент потерянных пакетов, процент искаженных пакетов

**44. Базовая архитектура QoS**

Слово «Архитектура» не встречается ни в пятой, ни в шестой лекциях, которые уделены изучению QoS. QoS упоминается в десятой лекции, но и там не даётся описание базовой архитектуры. Вероятнее всего, исходя из номера вопроса, имелись ввиду основные механизмы QoS на сетевом узле.

**Роль узловых средств QoS**

Средства QoS узла являются основным исполнительным механизмом службы QoS, так как именно они непосредственно влияют на процесс продвижения пакетов между входными и выходными интерфейсами коммутаторов и маршрутизаторов и, следовательно, определяют вклад данного устройства в характеристики качества обслуживания сети.

**Два типа механизмов QoS:**

**1. Обслуживание очередей**

Обязательный элемент любой пакетной коммутации.

- Алгоритмы обработки: от простого **FIFO** до сложных (приоритетное, взвешенное обслуживание).
- Алгоритм FIFO используется в сетевых устройствах по, но достаточен лишь для обслуживания типа «Best Effort». Для реализации настоящего QoS требуются более сложные механизмы.

**2. Кондиционирование трафика**

- Опциональный механизм.
- Решает задачу согласования скорости поступления и скорости продвижения трафика.

**45. Алгоритмы управления очередями**

Ответ нашёлся в шестой лекции.

**1) Алгоритм FIFO (First In – First Out).**

Преимущества:

- Простота реализации
- Не требует дополнительной настройки

Недостатки:

- Отсутствует дифференциация трафика
- Все пакеты обрабатываются одинаково
- Критичный трафик (голосовой) может задерживаться из-за фоновых задач (резервное копирование)

FIFO необходим для базовой работы сетевых устройств, но недостаточен для поддержки дифференцированного качества обслуживания.

**2. Алгоритм приоритетного обслуживания (Priority queuing).**

**Суть:** Алгоритм, который обрабатывает трафик не в порядке очереди, а на основе его важности (приоритета).

**Области применения:**

- Операционные системы (для приоритизации приложений).
- Сетевые устройства (для приоритизации классов трафика).

**Принцип:** Весь трафик делится на классы, каждому классу назначается числовой приоритет.

**Как это работает?**

- **Разделение на классы (Классификация):** Весь трафик делится на классы (по типу протокола, приложению, адресу).
- **Назначение приоритета:** Каждому классу назначается числовой приоритет (высокий, средний, низкий).
- **Маркировка:** Приоритет записывается в поле в заголовке пакета (например, IEEE 802.1Q/p).

**Цель:** Гарантировать минимальные задержки для критически важного трафика (голос, видео).

**Работа механизма очередей:**

**Принцип абсолютного приоритета:** Очереди обрабатываются строго от высшей к низшей.

- Сначала полностью обрабатывается очередь пакетов с высоким приоритетом.
- Только когда эта очередь становится пустой, обработка переходит к очереди со средним приоритетом и так далее.

**Управление буферами памяти:**

- Размер буфера для каждой очереди ограничен.
- Если буфер заполнен, новые пакеты этого класса теряются.
- Важному и интенсивному трафику можно выделить больший буфер.

**Преимущества (для высокоприоритетного трафика)**

- **Максимальная скорость:** Пакеты получают нужную пропускную способность.
- **Минимальные задержки:** Качество обслуживания (QoS) на высоком уровне.

**Недостатки (для низкоприоритетного трафика)**

- **Непредсказуемость:** Качество обслуживания может сильно падать.
- **Риск блокировки (заморозки):** Если высокоприоритетный трафик интенсивен, низкий может быть полностью остановлен.

**Применение:** приоритетное обслуживание идеально подходит для малоинтенсивного, но чувствительного к задержкам трафика (например, голос VoIP). Для интенсивного трафика (видео) нужны более сложные алгоритмы.

3. **Взвешенное обслуживание (Weighted Queuing)**

**Цель:** Гарантировать каждому классу трафика определенную долю пропускной способности.

**Суть:** Каждому классу назначается **вес** — процент от общей пропускной способности канала.

**Типы алгоритмов:**

- **Настраиваемая очередь (Custom Queuing - CQ):** Вес классов назначает администратор.
- **Взвешенное справедливое обслуживание (Weighted Fair Queuing - WFQ):** Автоматическое назначение весов.

**Принцип работы**

Трафик делится на классы, для каждого создается отдельная очередь пакетов.

Очереди обслуживаются циклически**.**

За один цикл из каждой очереди передается объем данных, пропорциональный ее весу.

**Механизм обслуживания очередей**

Циклический опрос всех очередей арбитром.

Ключевой параметр: Время цикла.

- Слишком большой цикл → высокие задержки.
- Слишком малый цикл → неэффективное использование CPU.

**Влияние на параметры QoS:**

1. **Пропускная способность:** Гарантирована в соответствии с весом.
2. **Задержки:** Зависят от:
   - Коэффициента нагрузки трафика класса.
   - Длины цикла обслуживания.
   - Интенсивности трафика всех классов.
3. **Потери пакетов:** Зависят от размера буфера, назначенного очереди.

**Преимущества:**

- **Справедливость:** Все классы получают гарантированную полосу.

- **Предсказуемость:** Избегается полная блокировка низкоприоритетных классов.
- **Гибкость:** Можно точно распределить ресурсы между классами.

**Недостатки:**

- **Более высокие задержки:** По сравнению с высокоприоритетной очередью.
- **Сложность прогнозирования:** Точные значения QoS зависят от многих факторов.
- **Требует настройки:** Необходимо правильно назначить веса и размеры буферов.

**Применение:** Когда нужно гарантировать минимальную полосу пропускания для всех критически важных классов трафика.

**4. Взвешенное справедливое обслуживание (WFQ)**

**Суть:** Гибридный алгоритм, сочетающий приоритетное и взвешенное обслуживание.

**Реализация:** Существует множество реализаций от разных производителей.

- Отличаются способами назначения весов и режимами работы.
- Важно изучать детали конкретной реализации.

**Два категории очередей.**

**1. Приоритетная очередь:**

- Обслуживается первой и полностью.
- Для критичного трафика: системные сообщения, управление, голос, видео.
- Предполагается низкая интенсивность, чтобы не блокировать остальной трафик.

**2. Набор взвешенных очередей:**

- Обслуживаются циклически при пустой приоритетной очереди.

Механизм распределения ресурсов:

**1. Приоритетная фаза:**

- Сначала вся полоса выделяется приоритетной очереди (пока она не опустеет).
- Затем оставшаяся полоса распределяется между взвешенными очередями.

**2. Взвешенная фаза:**

- **Вариант 1 (Ручной):** Администратор назначает веса (доли пропускной способности) для каждого класса.
- **Вариант 2 (По умолчанию):** Оставшаяся полоса делится поровну между всеми классами.

- **Результат:** Баланс между минимальными задержками для критичного трафика и гарантированной полосой для остальных.

46. Механизмы профилирования и формирования трафика

**1. Механизм и вариации алгоритма «Дырявое ведро» (Leaky Bucket)**

**Механизм алгоритма**

- Алгоритм использует счетчик (C), который представляет собой «ведро».
- При поступлении данных счетчик увеличивается.
- Каждые T секунд «ведро протекает»: значение счетчика уменьшается на величину **Bc**.

**Обработка кадров на основе состояния счетчика**

- **C ≤ Bc**: Кадр пропускается с DE=0.
- **Bc < C ≤ (Bc + Be)**: Кадр передается с признаком DE=1.
- **C > (Bc + Be)**: Кадр отбрасывается.

Алгоритм допускает различные модификации:

- Больше порогов для градации нарушений.
- Контроль пиковой скорости вместо объема пульсации.
- **GCRA (Generic Cell Rate Algorithm)** — модификация для сетей ATM, контролирующая пиковую и среднюю скорость, вариацию интервала прибытия ячеек и объем пульсации.

**2. Алгоритм «Ведро токенов» (Token Bucket)**

**Применение:** Профилирование и формирование (сглаживание) трафика.

**Цель:** Уменьшение неравномерности передачи пакетов, предотвращение их сбивания в плотные группы из-за пульсаций.

**Основная идея:** Пакеты передаются в сеть только при наличии разрешений (токенов), что обеспечивает равномерную отправку данных.

**Ключевые компоненты системы:**

- **Генератор токенов:** Создает токены с фиксированной скоростью **r** (бит/с).
- **Ведро:** Накопитель токенов с ограниченной емкостью **b** (бит).
- **Очередь пакетов:** Буфер для хранения поступающих пакетов объемом **K** бит.
- **Сервер:** Передает пакет в сеть, только если для него хватает токенов.

**Параметры алгоритма**

- **r** — Скорость генерации токенов (желательная средняя скорость трафика).
- **b** — Емкость ведра (максимальный объем запаса токенов).
- **m** — Объем одного токена (бит).

**Механизм работы:**

- **Накопление:** Токены поступают с фиксированной скоростью r и накапливаются в ведре (до предела b).
- **Проверка:** Каждый пакет требует для передачи токены в объеме своего размера.
- **Передача:** Если токенов достаточно → пакет передается, токены списываются. Если недостаточно → пакет ждет в очереди.

**Результат работы алгоритма:**

Сглаживание трафика — даже при поступлении пачки пакетов передача идет равномерно в темпе r.

Идеальный профиль — поток токенов задает эталонный трафик.

### 47. Протокол X.25

Информация взята из седьмой лекции.

**X.25** — стандарт передачи данных с использованием коммутации пакетов, разработанный в 1970-х годах для работы в глобальных сетях (WAN).

**Суть:** Протокол интерфейса между терминалом (DTE) и сетью (DCE).

Он обеспечивает надежную передачу данных даже по линиям связи низкого качества за счет активного контроля ошибок.

**Преимущества**:

- **Надежность**: Высокая устойчивость к помехам и ошибкам на линиях связи.
- **Управление соединением**: Возможность создания виртуальных каналов с управлением передачей данных.

**Недостатки**:

- **Задержки**: Большие задержки отклика, связанные с необходимостью коррекции ошибок (~0.6 сек).
- **Производительность**: Низкая производительность по сравнению с более современными протоколами.

### 48. Протоколы уровней X.25

Информация взята из седьмой лекции.

**Соответствует первым трем уровням модели OSI.**

**Уровень 1 (Физический):** определяет электронные и механические процедуры перевода в активное и пассивное состояние физической передающей среды, соединяющей устройства DTE и DCE.

**Уровень 2 (Канальный):** реализован с помощью протокола LAPB. Протокол LAPB обеспечивает фреймирование пакетов для каната DTE/DCE.

**Уровень 3 (Сетевой/Пакетный):** определяет форматы пакетов и процедуры обмена пакетами между одноранговыми объектами уровня 3.

Спецификации уровней 2 и 3 определены также стандартами IS0 7776 (протокол LAPB) и ISO 8208 (уровень пакетного обмена X.25).

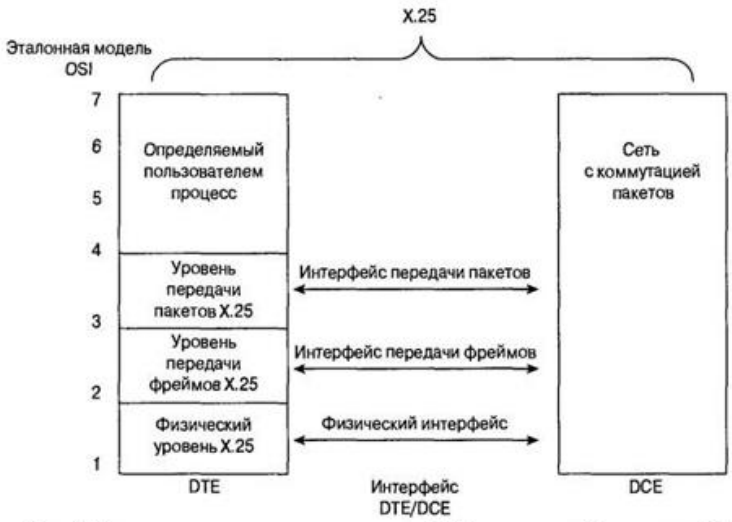



Рис.1. Соотношение между протоколом X.25 и эталонной моделью OSI.

### 49. Формат кадра X.25

Информация взята из седьмой лекции.

Структура кадра: Байта передаются, начиная с младшего бита.

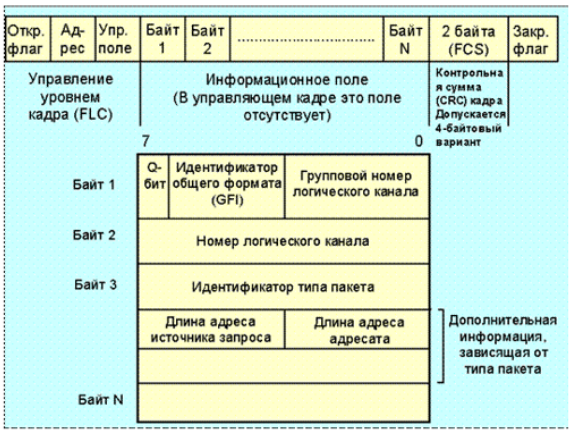



Рис.2. Формат кадра X.25.

| Поле | Размер | Описание |
|---|---|---|
| **Флаг** | 1 байт | 01111110 (0x7E). Маркер начала и конца кадра. |
| **Адрес** | 1 байт | Адрес получателя в сети. Ограничивает число устройств в одном канале. |
| **Управление** | 1 (или 2) байта | Служебная информация: тип кадра, номер последовательности, управление потоком. |
| **Данные** | Переменный | Пакет сетевого уровня (уровень 3) или управляющая информация. |
| **FCS (CRC)** | 2 байта | Контрольная последовательность для обнаружения ошибок в кадре. |
| **Флаг** | 1 байт | 01111110 (0x7E). Маркер конца кадра. |

Таблица.2. Визуальное представление формата кадра с указанием полей и их размеров.

### 50. Технология Frame Relay

**Технология Frame Relay (Коммутация кадров) –** это технология передачи данных для территориальных сетей с коммутацией пакетов.
Часто предоставляется операторами сетей X.25.

- **Ключевая аналогия:** Ethernet для глобальных сетей
  - Предоставляет быстрые базовые транспортные услуги.

    - Доставка кадров дейтаграммным способом без гарантий.
    - Сеть не восстанавливает потерянные или искаженные кадры.
- **Производительность** зависит от качества каналов и методов восстановления пакетов на уровнях стека, расположенного над протоколом Frame Relay.
    - Хорошие каналы: Высокая полезная пропускная способность (протоколы TCP/NCP справляются с редкими потерями).
    - Плохие каналы: Пропускная способность может упасть в десятки раз.
- **Ключевые ограничения технологии**
    - Нет гарантий на задержки → сложности с передачей голоса и видео.
    - Низкая скорость доступа (до 2 Мбит/с) → часто недостаточно для видео.
- **Требования к качеству каналов**
    - Магистральные каналы: только высококачественные волоконно-оптические.
    - Каналы доступа: уровень ошибок не ниже $10^{-6}$ (витая пара с G.703/ISDN допустима).
- **Передача голоса (с оговорками)**
    - Требует присвоения кадрам голоса высокого приоритета.
    - Сеть должна быть недогружена для минимизации задержек в очередях.
    - Необходимы кадры малого размера для снижения задержек пакетизации.
    - Спецификация FRF.11 — попытка стандартизации, но работа продолжается.
- **Основное применение**
    - Альтернатива выделенным линиям (T1/E1): Аналогичные услуги за меньшую плату.
    - Идеально для объединения локальных сетей через постоянные виртуальные каналы (PVC).
- **Оборудование доступа (FRAD)**
    - Удаленные мосты: не нуждаются в интеллекте для установления соединения.
    - Маршрутизаторы: наиболее частый вариант доступа с поддержкой Frame Relay.

**Стандарт RFC 1490** определяет инкапсуляцию кадров Ethernet/FDDI для передачи по Frame Relay.

**51. Назначение и общая характеристика сети Frame Relay**

- Основное назначение
    - Публичная сеть для эффективного соединения корпоративных локальных сетей (LAN).
    - Оптимизирована для передачи «пульсирующего» трафика локальных сетей.
- Ключевые преимущества перед X.25

    - Низкая протокольная избыточность → высокая пропускная способность и малые задержки.
    - Дейтаграммный режим работы (без гарантий доставки).
    - Поддержка гарантированной средней скорости передачи данных по виртуальному каналу при допустимых пульсациях трафика.
- Ограничения
    - Требует высококачественных каналов (оптоволокно).
    - Максимальная скорость доступа: 2 Мбит/с.
- Организации по стандартизации
    - ITU-T (CCITT): Основные международные стандарты.
    - ANSI T1S1: Национальный американский институт стандартов.
    - Frame Relay Forum (FRF): Консорциум ведущих производителей (Cisco, Northern Telecom и Digital Equipment Corporation).
- Роль Frame Relay Forum
    - Разработка практических спецификаций (FRF.X).
    - Введение дополнительных возможностей, например:
        - LMI (Local Management Interface) для управления сетью.
        - FRF.11 для передачи голоса.
    - Упрощение и адаптация сложных стандартов ITU-T.
- Типы виртуальных каналов
    - PVC (Постоянный виртуальный канал):
        - Постоянное соединение.
        - Исторически первая и до сих пор основная услуга.
    - SVC (Коммутируемый виртуальный канал):
        - Устанавливается по требованию.
        - Поддержка появилась позже, но доступна у многих поставщиков.
- Эволюция предложений
    - Изначально поставщики предлагали только PVC (как упрощение технологии).
    - Со временем появилась поддержка SVC, дающая гибкость для временных соединений.

**52. Стек протоколов Frame Relay**

Технология Frame Relay использует для передачи данных технику виртуальных соединений, аналогичную той, которая применялась в сетях X.25.

Технология Frame Relay работает на **физическом и канальном** уровнях (X.25 — ещё и на сетевом).

**Основной протокол:** LAP-F (Link Access Protocol for Frame Relay).

Протокол канального уровня LAP-F в сетях Frame Relay имеет два режима работы:

- **Основной (core) :** Фактически используемый режим. Передача без контроля и преобразований.
- **Управляющий (control) :** Необязательная надстройка. Для контроля доставки и управления потоком (Frame Switching).

**Преимущества основного режима**

- Низкие накладные расходы.
- Минимальные задержки.
- Пакеты локальных сетей инкапсулируются напрямую в кадры Frame Relay.

**Для коммутируемых каналов (SVC)** используется канал D и стек протоколов ISDN:

- **LAP-D**: Обеспечивает надёжную передачу управляющих кадров.
- **Q.931/Q.933**: Устанавливает виртуальное соединение.

**После установки** соединения данные передаются по протоколу **LAP-F Core**.

**2. Идентификатор соединения.** Для идентификации соединения используется DLCI (Data Link Connection Identifier), который является аналогом номера виртуального канала.

**3. При использовании PVC** оборудованию Frame Relay нужно поддерживать только протокол LAP-F core.

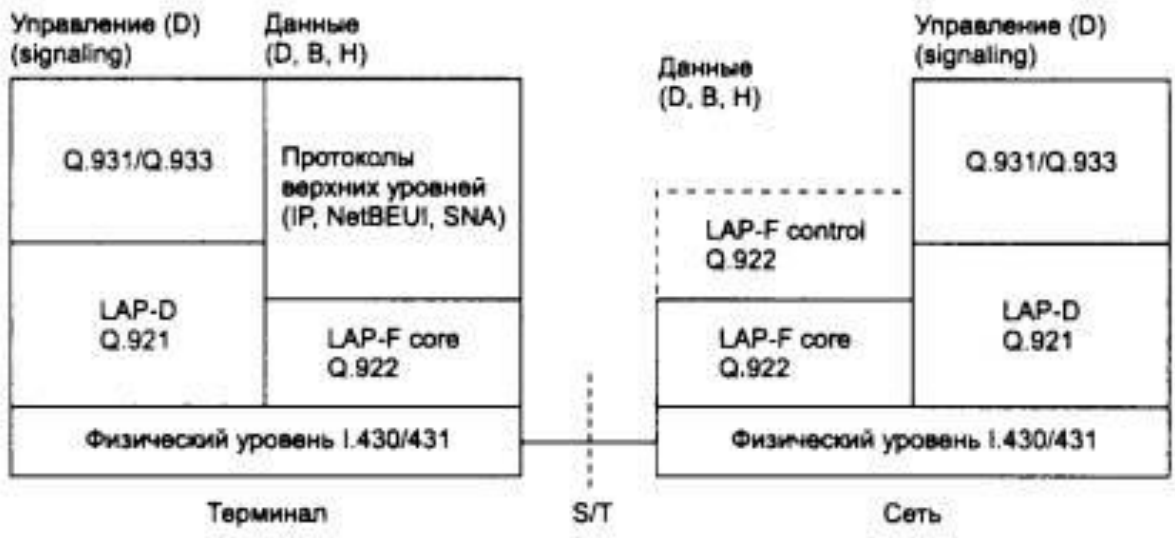


Рис. 1. Стек протоколов Frame Relay.

- **Структура кадра (на основе HDLC).**
- **Ключевые поля адреса:**
    - DLCI (Data Link Connection Identifier): идентификатор виртуального соединения (10 бит → 1024 соединения).
    - FECN, BECN, DE: для управления трафиком и поддержания заданного качества обслуживания виртуального канала.
    - C/R: признак «команда-ответ».
    - EAO и EA1 (Extended Address - Расширенный адрес).
- **Обработка ошибок:**
    - Технология **не исправляет** ошибки, а отбрасывает искажённые кадры.
    - Коррекция ошибок — задача вышележащих протоколов (например, TCP).
- Стандарты Frame Relay (ANSI, ITU-T) распределяют адреса DLCI между пользователями и сетью следующим образом:

| Диапазон DLCI | Назначение |
|---|---|
| 0 | Используется для виртуального канала локального управления (LMI) |
| 1 -15 | Зарезервированы для дальнейшего применения |
| 16-991 | Используются абонентами для нумерации PVC и SVC |
| 992-1007 | Используются сетевой транспортной службой для внутрисетевых соединений |
| 1008-1022 | Зарезервированы для дальнейшего применения |
| 1023 | Используются для управления канальным уровнем |

В любом интерфейсе Frame Relay для оконечных устройств пользователя отводится 976 адресов DLCI.

## 53. Технология ATM

- **Цель создания:** Компромисс между простотой коммутации и эффективным использованием ресурсов сети.
- **Что такое ATM?**
- **ATM (Asynchronous Transfer Mode — Асинхронной режим передачи)** – Сетевая высокопроизводительная технология коммутации и мультиплексирования пакетов.
- **Ключевая концепция:** Обеспечение временной прозрачности (постоянной скорости передачи).

- **Как это работает?**
- Вся информация разбивается на маленькие ячейки фиксированной длины.
- **Размер ячейки:** 53 байта (5 байт — заголовок, 48 байт — данные).
- Ячейки от разных источников асинхронно мультиплексируются в единый цифровой тракт.
- Управление трафиком с помощью приоритетов.
- **Высший приоритет:** Ячейки с данными, критичными ко времени (голос, видео).
    - Передаются в первую очередь.
- **Низший приоритет:** Ячейки с данными, не критичными ко времени (файлы).
    - Передаются в паузах между высокоприоритетным трафиком.
- **Преимущество:** Малая длина ячейки позволяет одновременно передавать несколько потоков с гарантированной скоростью.
- **Проблема внедрения:**
- Положительные стороны ATM затрудняют её продвижение.
- Сложность адаптации существующего оконечного оборудования к режиму ATM.

**54. Основные характеристики классов трафика ATM**

- ATM определяет классы трафика для гарантии QoS.

| Класс трафика | Характеристики | Примеры |
|---|---|---|
| A | CBR, синхронизация, с соединением | Голос, видео |
| B | VBR, синхронизация, с соединением | Сжатое голос/видео |
| C | VBR, без синхронизации, с соединением | TCP, Frame Relay, X.25, LLC2 |
| D | VBR, без синхронизации, без соединения | IP, Ethernet, DNS, SNMP |
| X | Тип трафика и его параметры определяются пользователем | - |

- В технологии ATM для каждого класса трафика определен набор количественных параметров, которые приложение должно задать.
- **Параметры трафика (пропускная способность):**
    - **PCR (Peak Cell Rate):** Максимальная скорость передачи данных.

- **SCR (Sustained Cell Rate):** Средняя скорость передачи данных.
- **MCR (Minimum Cell Rate):** Минимальная скорость передачи данных.
- **MBS (Maximum Burst Size):** Максимальный размер пульсации.

- **Параметры QoS (качество):**
  - **CTD (Cell Transfer Delay):** Задержка передачи ячеек.
  - **CDV (Cell Delay Variation):** Вариация задержки ячеек.
  - **CLR (Cell Loss Ratio):** Доля потерянных ячеек.

**Управление трафиком**

**1. Управление перегрузками**

- При насыщении пропускной способности ATM **отбрасывает ячейки** для минимизации задержек.
- **Стратегия отбрасывания:**
  - В первую очередь отбрасываются ячейки с **низким приоритетом** (например, данные).
  - Продвинутые коммутаторы могут отбрасывать **целые пакеты** для снижения нагрузки и избыточных повторных передач.
  - Правила отбрасывания ячеек определяются **QoS**.

**2. Трафик-контракт**

- Соглашение между приложением и сетью ATM о параметрах обслуживания.
- **Отличия от Frame Relay:**
  - Несколько классов трафика (A, B, C, D, X) с разными требованиями к задержкам и надежности.
  - Возможность задания не только пропускной способности, но и параметров QoS (задержка, потери).

**3. Дополнительные классы обслуживания.**

- **Класс X:** Для трафика со специфическими требованиями, не укладывающимися в стандартные классы. Параметры задаются вручную.
- **UBR (Unspecified Bit Rate):** Сервис «Best Effort». Применяется, когда параметры пропускной способности и QoS некритичны. Сеть предоставляет ресурсы по остаточному принципу.

**4. Поддержка виртуальных каналов**

- ATM поддерживает как постоянные (PVC), так и коммутируемые (SVC) виртуальные каналы.

- Автоматическое заключение трафик-контракта при установлении коммутируемого виртуального соединения представляет собой весьма непростую задачу. Требует от коммутаторов проверки доступности ресурсов.

**55. Архитектура ATM**

- Модель ATM, в соответствии с определением ANSI, ITU и ATM Forum, состоит из трех уровней:
    - физического;
    - уровня ATM;
    - уровня адаптации ATM (AAL).
- Стек протоколов ATM соответствует нижним уровням семиуровневой модели OSI. Прямого соответствия между уровнями протоколов технологии ATM и уровнями модели OSI нет.

**Структура стека протоколов ATM**

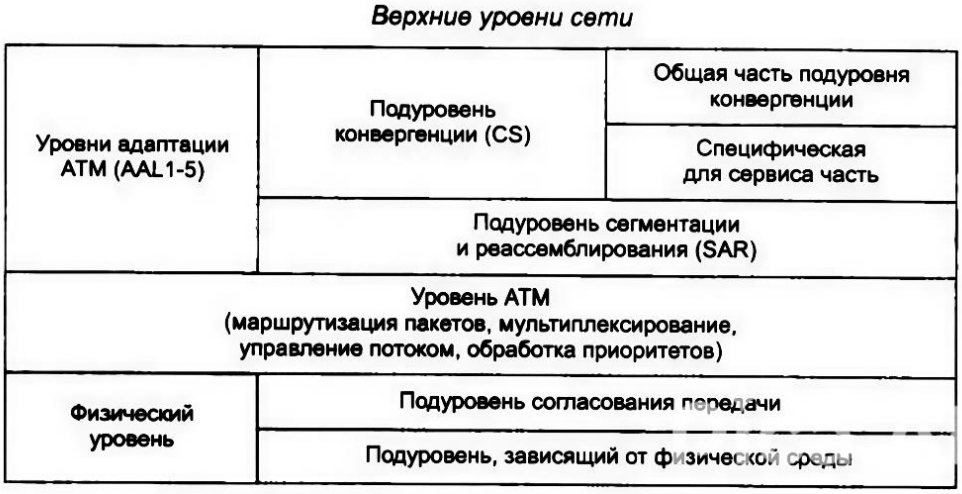



Рис. 2. Структура стека протоколов ATM.

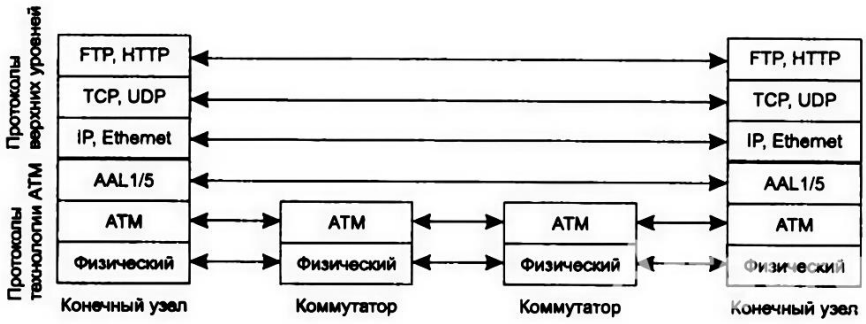



Рис. 3. Распределение протоколов по узлам и коммутаторам ATM.

**56. Формат ячейки ATM**

- **Ключевые поля заголовка (5 байт):**

1. **Управление потоком (Generic Flow Control - GFC):** Только между узлом и первым коммутатором (функции не стандартизированы).
2. **Идентификатор виртуального пути/канала (Virtual Path/Cannel Identifier - VPI/VCI):** Эти поля задают номер виртуального соединения, разделенный на старшую (VPI - 8 бит) и младшую (VCI - 16 бит) части для коммутации.
3. **Идентификатор типа данных (Payload Type Identifier - PTI):** Состоит из 3-х бит и задает:
    - Тип данных (пользовательские/управляющие).
    - Индикатор перегрузки в сети (Explicit Congestion Forward Identifier - EFCI).
4. **Приоритет потери пакета (Cell Loss Priority - CLP ):** Приоритет потери ячейки (1 - низкий, 0 - высокий). Аналог DE в Frame Relay.
5. **Управление ошибками в заголовке (Header Error Control - HEC):** Контрольная сумма для обнаружения/ исправления ошибок в заголовке и поиска границ ячеек.

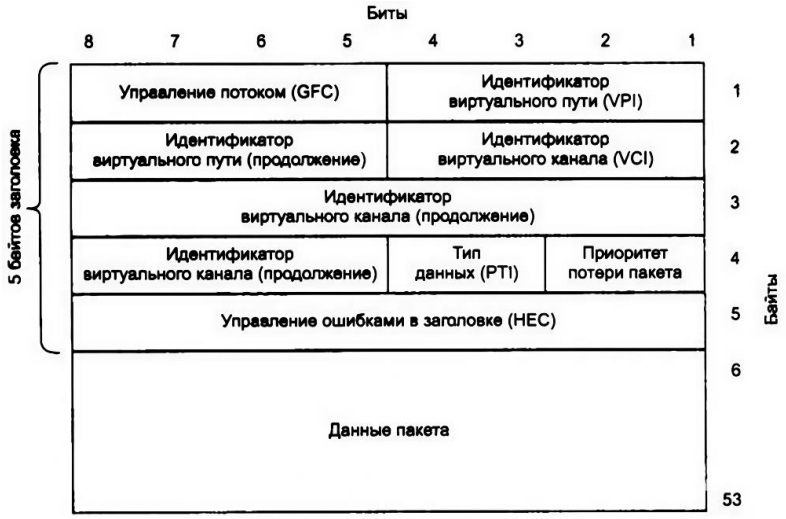



Рис.1. Формат ячейки ATM.