Инфокоммуникационные системы и сети

Организация сети.
Эталонная модель OSI.

Понятие сети

- **Сеть** (network) – группа компьютеров, принтеров, маршрутизаторов, сетевых устройств, которые обмениваются информацией через некоторую среду передачи данных.
  - **Локальные сети** (Local Area Networks) – позволяют объединить информационные ресурсы предприятия (файлы, принтеры, БД) для повышения эффективности применения вычислительной техники.
  - **Городские сети** (Metropolitan Area Networks) – позволяют организовать обмен данными между отдельными компьютерами и сетями отдельного региона, города.
  - **Глобальные сети** (Wide Area Networks) – позволяют обмениваться данными между отдельными сетями предприятий, удаленными на значительные расстояния.

Организация сети

- Организацией сети называется обеспечение взаимодействия между рабочими станциями, периферийным оборудованием и другими устройствами.
- Важной задачей является согласование различных типов компьютеров (Macintosh, IBM-совместимые, мэйнфреймы).
- Для обеспечения совместимости обмена данными используются *сетевые протоколы* – формальное описание набора правил о том, как устройства выполняют обмен информацией.

Преимущества сетевых технологий

- Первые вычислительные системы представляли собой *автономные системы*.
- Для повышения эффективности использования компьютерных систем используется их объединение в вычислительную сеть.
- Такой подход позволяет:
  - Устранить дублирование оборудование и ресурсов;
  - Обеспечить эффективный обмен данными между устройствами;
  - Обеспечить разделение процессов хранения и обработки информации.

Локальные сети

- Локальные сети служат для объединения рабочих станций, периферийных устройств, терминалов и других устройств.
- Характерные особенности локальной сети:
  - Ограниченные географические пределы;
  - Обеспечение пользователям доступа к среде с высокой пропускной способностью;
  - Постоянное подключение к локальным сервисам;
  - Физическое соединение рядом стоящих устройств.

Глобальные сети

- Глобальные сети служат для объединения локальных сетей и обеспечивают связь между компьютерами, находящимися в локальных сетях.
- Глобальные сети охватывают значительные географические пространства и обеспечивают возможность связать устройства на большом удалении друг от друга.

Сетевые стандарты

- Для решения проблемы совместимости различных систем *Международная организация по стандартизации (International Organization for Standardization, ISO)* в 1984 году выпустила эталонную модель взаимодействия открытых систем (OSI).
- Эталонная модель OSI является основной архитектурной моделью взаимодействия между компьютерами.

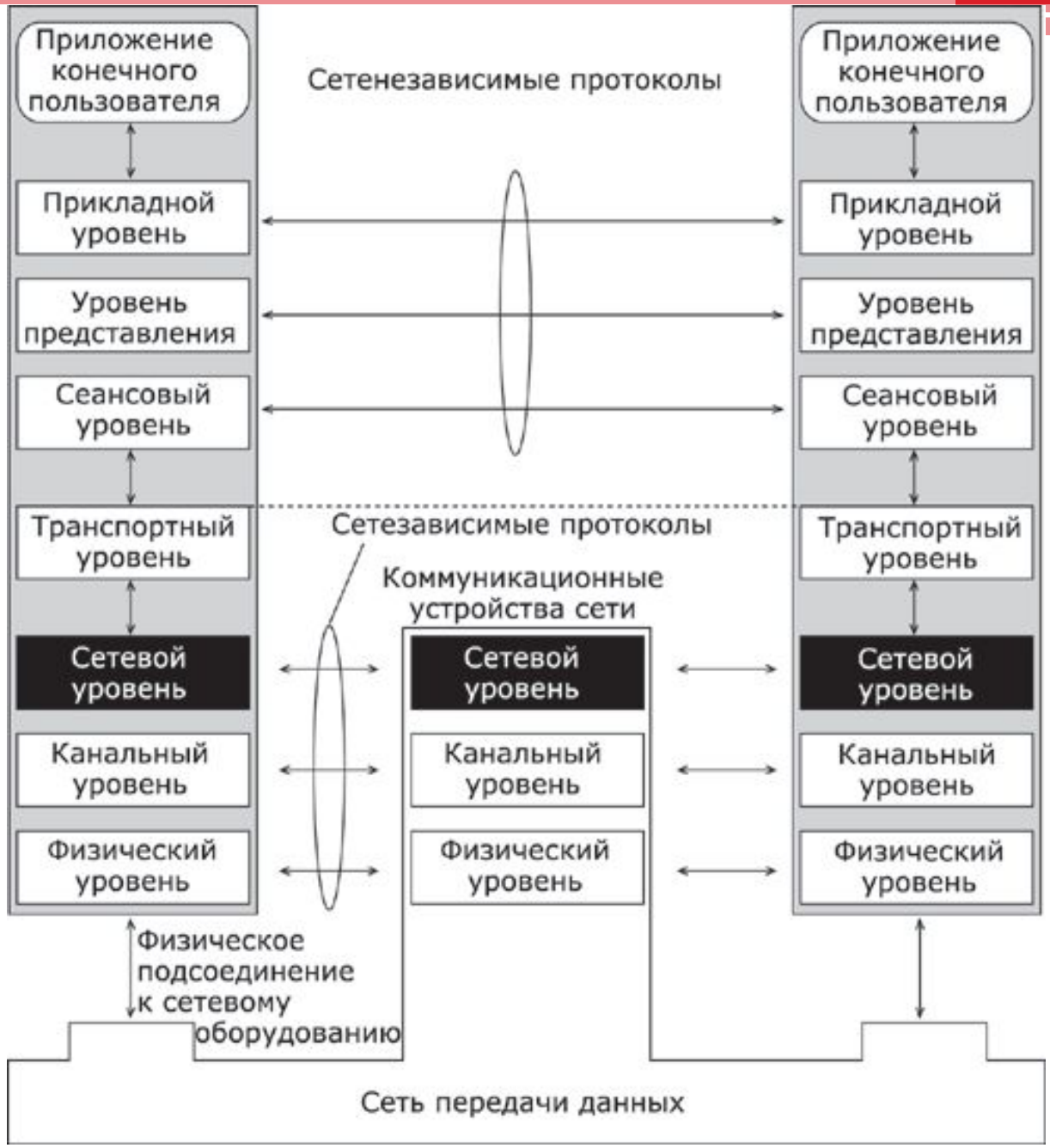
Модель взаимодействия открытых систем (OSI)

- Эталонная модель OSI – концептуальная схема сети, определяющая сетевые функции, реализуемые на каждом уровне.
- Эталонная модель OSI делит задачу перемещения информации между компьютерами через сетевую среду на семь подзадач.
  - Разделение на уровни называется *иерархическим представлением.*

Семь уровней эталонной модели OSI

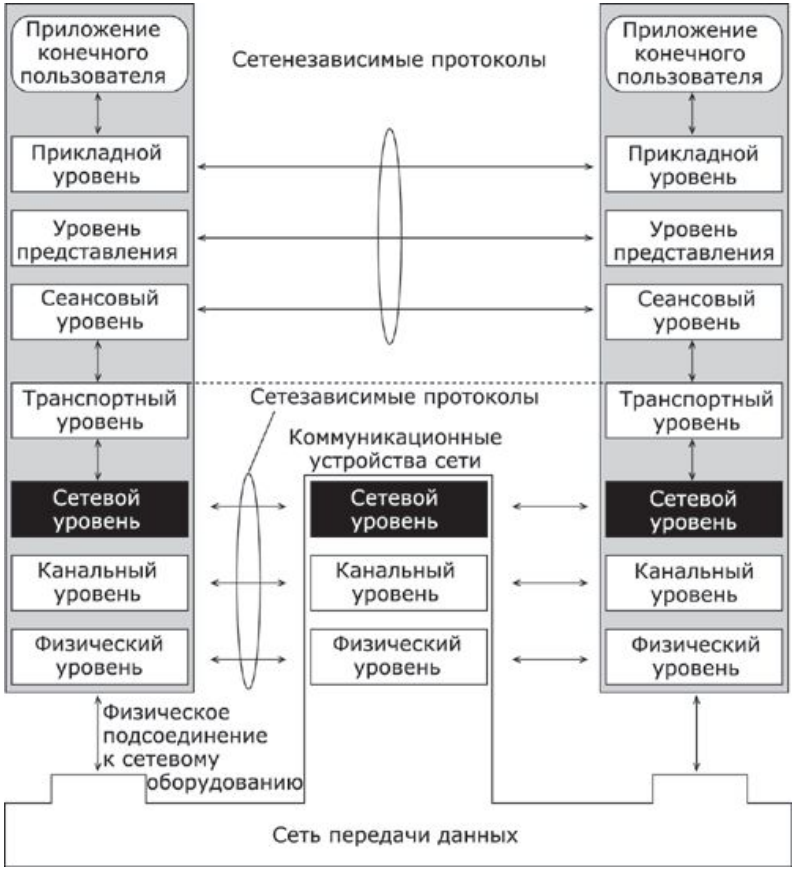
7	Уровень приложений	→	Сетевые процессы с прикладными программами
6	Уровень представлений	→	Представление данных
5	Сеансовый уровень	→	Связь между хостами
4	Транспортный уровень	→	Связь между конечными устройствами
3	Сетевой уровень	→	Адрес и маршрутизация
2	Канальный уровень	→	Доступ к среде передачи данных
1	Физический уровень	→	Двоичная передача

Модель OSI


Приложение конечного пользователя
Сетенезависимые протоколы
Приложение конечного пользователя
Прикладной уровень
Прикладной уровень
Уровень представления
Уровень представления
Сеансовый уровень
Сеансовый уровень
Транспортный уровень
Сетезависимые протоколы
Транспортный уровень
Коммуникационные устройства сети
Сетевой уровень
Сетевой уровень
Сетевой уровень
Канальный уровень
Канальный уровень
Канальный уровень
Физический уровень
Физический уровень
Физический уровень
Физическое подсоединение к сетевому оборудованию
Сеть передачи данных

Модель OSI

- Нижние уровни (с 1 по 3) модели OSI управляют физической доставкой сообщений и их называют *уровнями среды передачи данных (media layers)*.
- Верхние уровни (с 4 по 7) модели OSI призваны обеспечить точную доставку данных между компьютерами в сети и их называют *уровнями хост-машины (host layers)*.
- Модель OSI *не является* схемой реализации сети, она только определяет функции каждого уровня.



Цель разработки эталонной модели

- Деление функциональных задач сети на семь уровней в рамках модели OSI обеспечивает следующие преимущества:
  - Делит аспекты межсетевого взаимодействия на ряд менее сложных элементов;
  - Определяет стандартные интерфейсы для автоматического интегрирования в систему новых устройств и обеспечения совместимости сетевых продуктов разных поставщиков;
  - Позволяет закладывать в различные модульные функции межсетевого взаимодействия симметрию;
  - Изменения в одной области не требует изменений в других областях;
  - Делит сложную межсетевую структуру на дискретные, более простые для изучения подмножества операций.

Семь уровней эталонной модели OSI

- Уровень 7 (уровень приложений)
  - Уровень приложений – самый близкий к пользователю уровень модели OSI. Данный уровень не предоставляет услуги другим уровням, а только обслуживает прикладные процессы вне пределов модели OSI.
  - Уровень приложений идентифицирует и устанавливает доступность предлагаемых партнеров для связи, синхронизирует совместно работающие прикладные программы, а также устанавливает договоренности о процедурах восстановления после ошибок и контроля целостности данных.

Семь уровней эталонной модели OSI

- Уровень 6 (уровень представлений)
  - Уровень представлений отвечает за то, чтобы информация, посылаемая из уровня приложений одной системы, была читаемой для уровня приложений другой системы.
  - При необходимости уровень представлений преобразовывает форматы данных путем использования общего формата представления информации.

Семь уровней эталонной модели OSI

- Уровень 5 (сеансовый)
  - Сеансовый уровень устанавливает, управляет и завершает сеансы взаимодействия приложений.
  - Сеансы включают диалог между двумя или более объектами представления. Сеансовый уровень синхронизирует диалог между объектами уровня представлений и управляет обменом информации между ними.
  - Сеансовый уровень обеспечивает класс услуг и средства формирования отчетов для формирования отчетов об особых ситуациях.

Семь уровней эталонной модели OSI

- Уровень 4 (транспортный)
  - Транспортный уровень сегментирует и повторно собирает данные в один поток.
  - Транспортный уровень обеспечивает транспортировку данных, изолируя верхние уровни от деталей ее реализации.
  - Транспортный уровень обеспечивает механизмы для установки, поддержания и упорядоченного завершения действий виртуальных каналов, обнаружения и устранения неисправностей транспортировки, а также управления информационным потоком.

Семь уровней эталонной модели OSI

- Уровень 3 (сетевой)
  - Сетевой уровень – комплексный уровень, обеспечивающий соединение и выбор маршрута между конечными системами, которые могут располагаться географически в разных сетях.
- Уровень 2 (канальный)
  - Канальный уровень обеспечивает надежный транзит данных через физический канал.
  - Канальный уровень решает вопросы физической адресации, топологии сети, уведомления об ошибках, упорядоченной доставки кадров, а также управления потоком данных.

Семь уровней эталонной модели OSI

- Уровень 1 (физический)
  - Физический уровень определяет электротехнические, механические, процедурные и функциональные характеристики активизации, поддержания и деактивизации физического канала между конечными системами.
  - Спецификации физического уровня определяют такие характеристики, как уровни напряжений, временные параметры изменения напряжений, скорости физической передачи данных и т.п.

Одноранговая модель взаимодействия

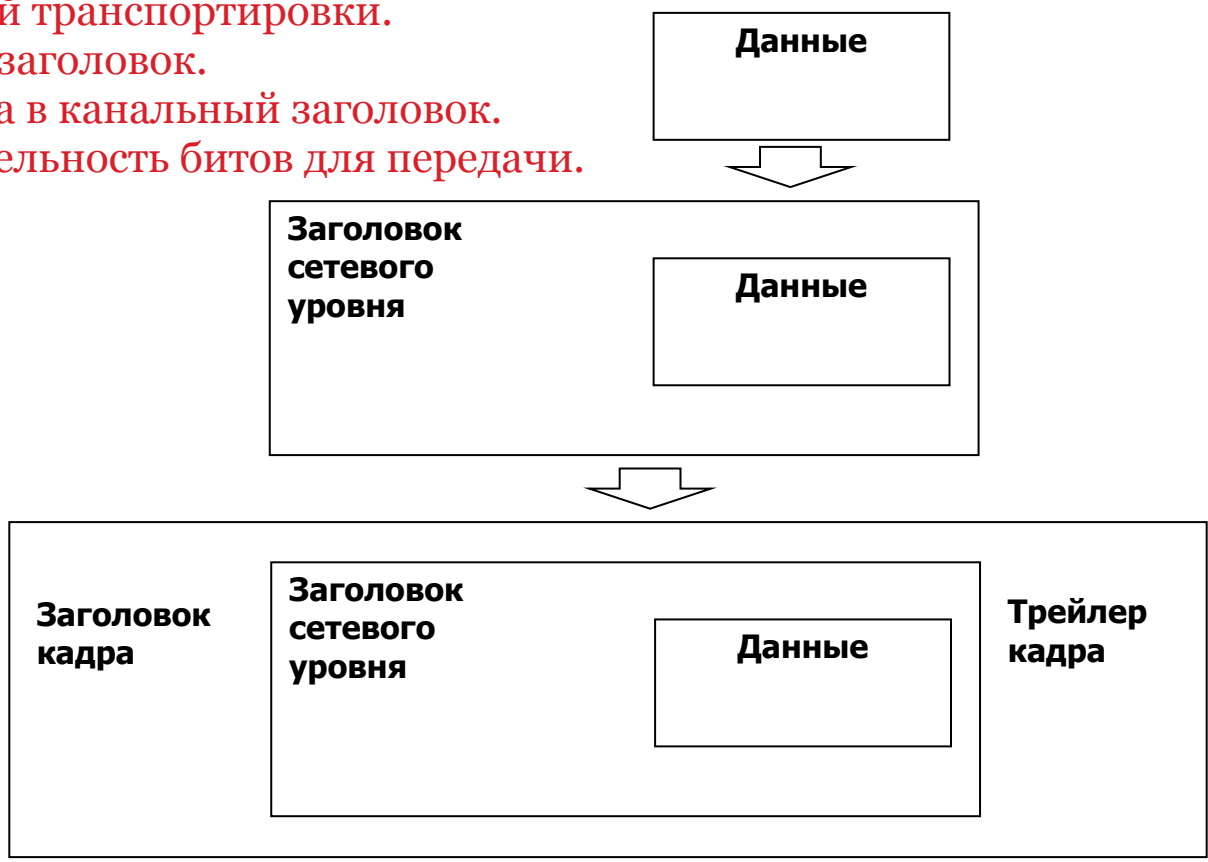
- Многоуровневая модель OSI исключает прямую связь между равными по положению уровнями в разных компьютерных системах.
- Каждый уровень решает свои задачи. Для выполнения своих задач, он должен общаться с соответствующим уровнем в другой системе.
- Обмен сообщениями (*блоками данных протокола – protocol data units, PDU)* осуществляется с помощью протокола соответствующего уровня.
- Обмен данными достигается за счет использования услуг уровней, лежащих на более низких уровнях.

Инкапсуляция данных

- Информация, посланная в сеть, называется данными или пакетами данных.
  - Если один компьютер (источник) посылает данные другому компьютеру (получателю), то данные должны быть собраны в пакет в процессе инкапсуляции.
- Каждый уровень эталонной модели зависит от услуг нижележащего уровня.
  - Нижний уровень при помощи инкапсуляции помещает блок PDU, полученный от верхнего уровня, в свое поле данных.
  - Затем добавляются заголовки и трейлеры, необходимые уровню для реализации своей функции.

Инкапсуляция данных

- Процесс передачи данных может быть схематично представлен следующим образом:
  - Формирование данных.
  - Упаковка данных для сквозной транспортировки.
  - Добавление сетевого адреса в заголовок.
  - Добавление локального адреса в канальный заголовок.
  - Преобразование в последовательность битов для передачи.



Взаимодействие в сети

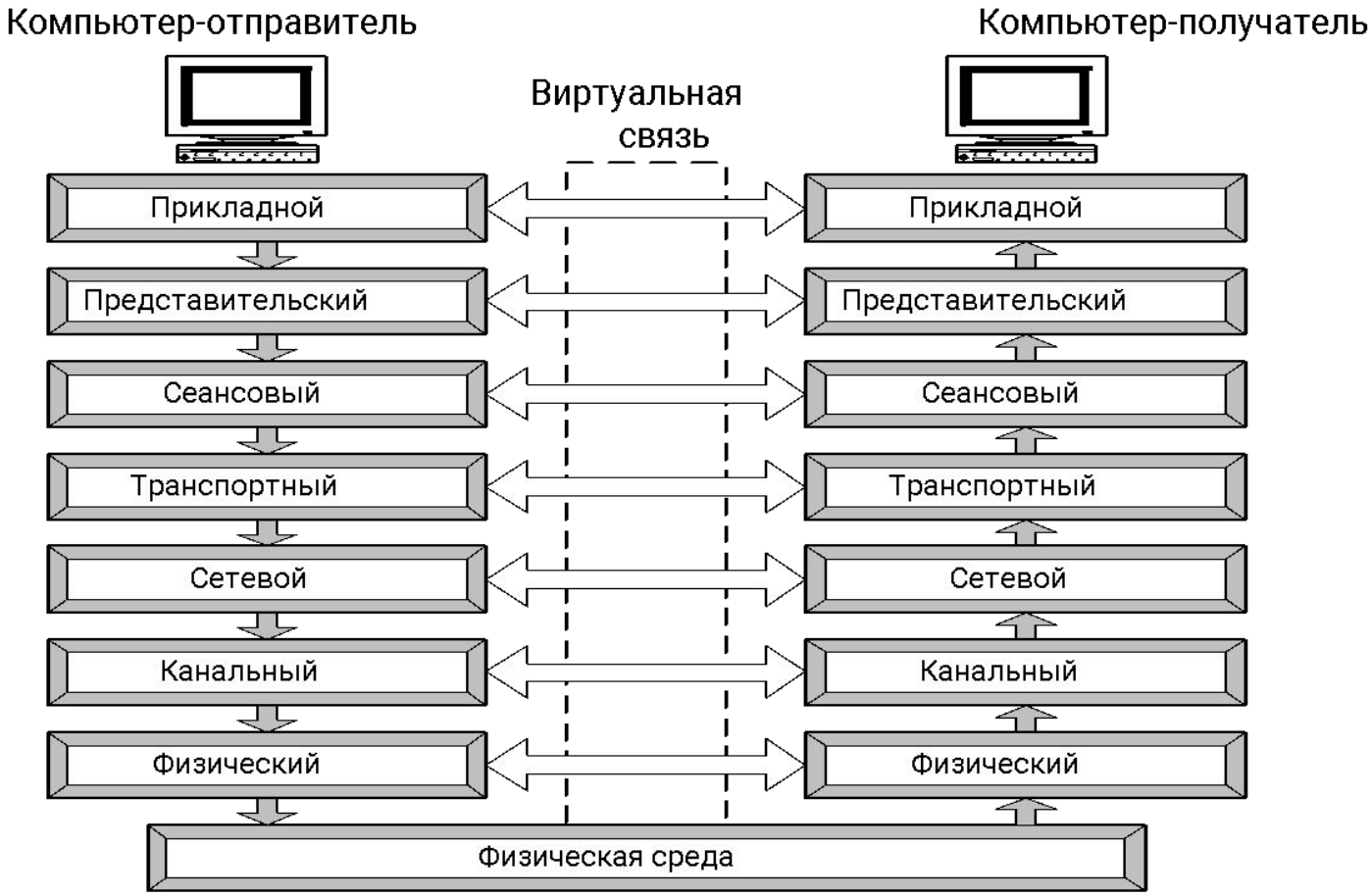
Взаимодействие в сети

- Каждый уровень на одном абоненте работает так, как будто он имеет прямую связь с соответствующим уровнем другого абонента.
- Между одноименными уровнями абонентов сети существует виртуальная (логическая) связь, например, между прикладными уровнями взаимодействующих по сети абонентов.
- Реальную физическую связь (кабель, радиоканал) абоненты одной сети имеют только на самом нижнем, первом, физическом уровне.
- В передающем абоненте информация проходит все уровни, начиная с верхнего и заканчивая нижним. В принимающем абоненте полученная информация совершает обратный путь: от нижнего уровня к верхнему.

Взаимодействие уровней модели OSI

- Модель OSI можно разделить на две различных модели:
- горизонтальную модель на базе протоколов, обеспечивающую механизм взаимодействия программ и процессов на различных машинах;
- вертикальную модель на основе услуг, обеспечиваемых соседними уровнями друг другу на одной машине.

Схема взаимодействия компьютеров в базовой эталонной модели OSI



- Каждый уровень компьютера–отправителя взаимодействует с таким же уровнем компьютера-получателя, как будто он связан напрямую. Такая связь называется логической или виртуальной связью. В действительности взаимодействие осуществляется между смежными уровнями одного компьютера.

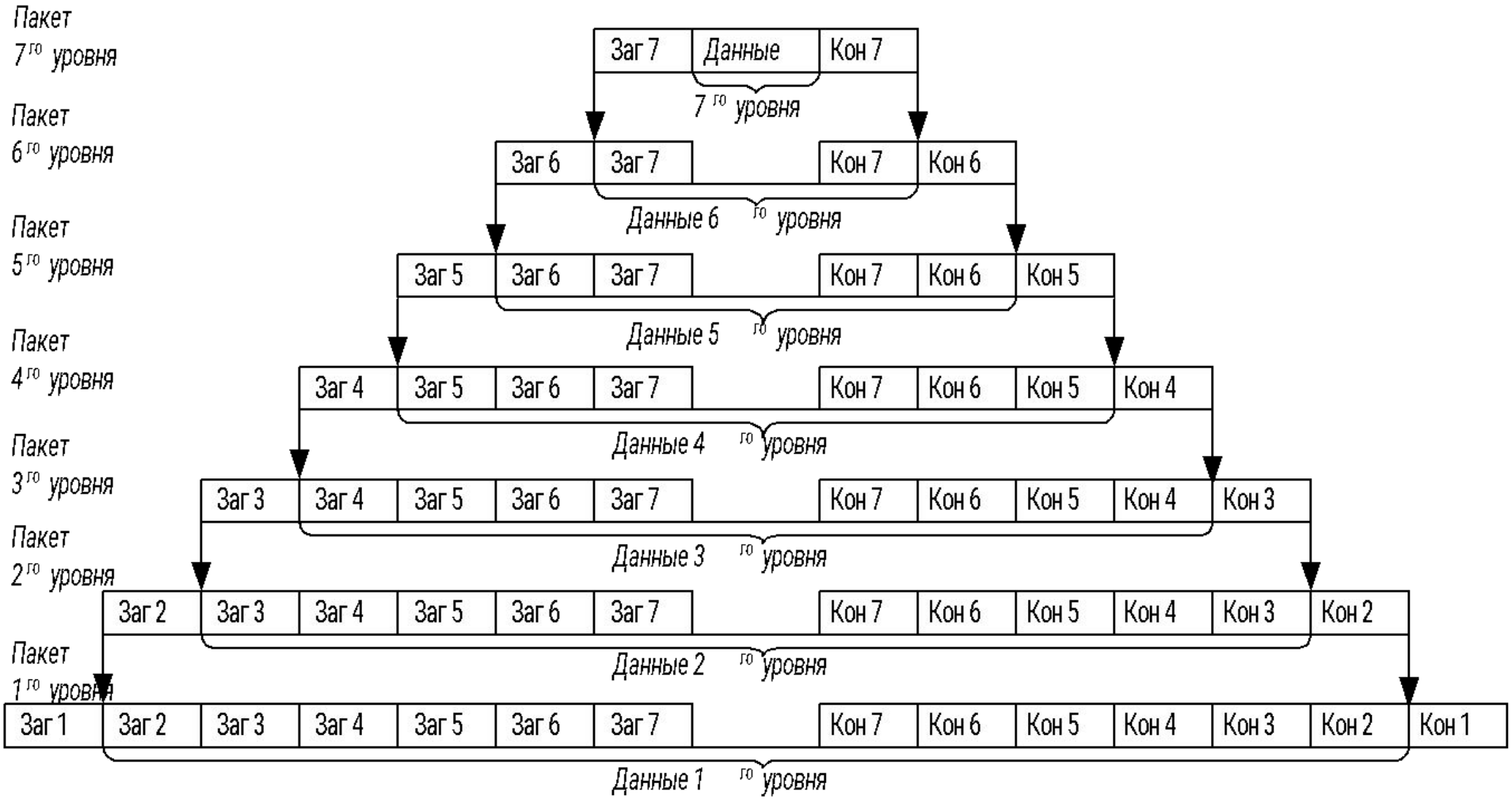
- информация на компьютере-отправителе должна пройти через все уровни. Затем она передается по физической среде до компьютера–получателя и опять проходит сквозь все слои, пока не доходит до того же уровня, с которого она была послана на компьютере-отправителе.

- В горизонтальной модели двум программам требуется общий протокол для обмена данными. В вертикальной модели соседние уровни обмениваются данными с использованием интерфейсов прикладных программ API (Application Programming Interface).

- Перед подачей в сеть данные разбиваются на пакеты. Пакет (packet) – это единица информации, передаваемая между станциями сети. При отправке данных пакет проходит последовательно через все уровни программного обеспечения. На каждом уровне к пакету добавляется управляющая информация данного уровня (заголовок), которая необходима для успешной передачи данных по сети, как это показано на рис. где *Заг* – заголовок пакета, *Кон* – конец пакета.

- На принимающей стороне пакет проходит через все уровни в обратном порядке. На каждом уровне протокол этого уровня читает информацию пакета, затем удаляет информацию, добавленную к пакету на этом же уровне отправляющей стороной, и передает пакет следующему уровню. Когда пакет дойдет до *Прикладного* уровня, вся управляющая информация будет удалена из пакета, и данные примут свой первоначальный вид.

Формирование пакета каждого уровня семиуровневой модели



- Каждый уровень модели выполняет свою функцию. Чем выше уровень, тем более сложную задачу он решает
- Каждый уровень обеспечивает сервис для вышестоящего уровня, запрашивая в свою очередь, сервис у нижестоящего уровня. Верхние уровни запрашивают сервис почти одинаково: как правило, это требование маршрутизации каких-то данных из одной сети в другую. Практическая реализация принципов адресации данных возложена на нижние уровни.

Рассматриваемая модель определяет взаимодействие открытых систем разных производителей в одной сети. Поэтому она выполняет для них координирующие действия по:

- взаимодействию прикладных процессов;
- формам представления данных;
- единообразному хранению данных;
- управлению сетевыми ресурсами;
- безопасности данных и защите информации;
- диагностике программ и технических средств.

краткое описание функций всех уровней

7. Прикладной представляет набор интерфейсов, позволяющий получить доступ сетевым службам
6. Представления преобразует данные в общий формат для передачи по сети
5. Сеансовый поддержка взаимодействия (сеанса) между удаленными процесс
4. Транспортный управляет передачей данных по сети, обеспечивает подтвержден передачи
3. Сетевой маршрутизация, управление потоками данных, адресация сообще для доставки, преобразование логические сетевые адреса и имен соответствующие им физические
2. Канальный **2.1. Контроль логической связи (LLC):** формирование кадров **2.2. Контроль доступа к среде (MAC):** управление доступом к среде
1. Физический: битовые протоколы передачи информации

Прикладной уровень выполняет следующие функции:

Описание форм и методов взаимодействия прикладных процессов.

1. Выполнение различных видов работ.

   передача файлов;

   управление заданиями;

   управление системой и т.д.

1. Идентификация пользователей по их паролям, адресам, электронным подписям;
2. Определение функционирующих абонентов и возможности доступа к новым прикладным процессам;
3. Определение достаточности имеющихся ресурсов;
4. Организация запросов на соединение с другими прикладными процессами;
5. Передача заявок представительскому уровню на необходимые методы описания информации;

6. Выбор процедур планируемого диалога процессов;
7. Управление данными, которыми обмениваются прикладные процессы и синхронизация взаимодействия прикладных процессов;
8. Определение качества обслуживания (время доставки блоков данных, допустимой частоты ошибок);
9. Соглашение об исправлении ошибок и определении достоверности данных;
10. Согласование ограничений, накладываемых на синтаксис (наборы символов, структура данных).

- Указанные функции определяют виды сервиса, которые прикладной уровень предоставляет прикладным процессам. Кроме этого, прикладной уровень передает прикладным процессам сервис, предоставляемый физическим, канальным, сетевым, транспортным, сеансовым и представительским уровнями.

- На *прикладном уровне* необходимо предоставить в распоряжение пользователей уже переработанную информацию. С этим может справиться системное и пользовательское программное обеспечение.
- Прикладной уровень отвечает за доступ приложений в сеть. Задачами этого уровня является перенос файлов, обмен почтовыми сообщениями и управление сетью

- К числу наиболее распространенных протоколов верхних трех уровней относятся:
- FTP (File Transfer Protocol) протокол передачи файлов;
- TFTP (Trivial File Transfer Protocol) простейший протокол пересылки файлов;
- X.400 электронная почта;
- Telnet работа с удаленным терминалом;
- SMTP (Simple Mail Transfer Protocol) простой протокол почтового обмена;
- CMIP (Common Management Information Protocol) общий протокол управления информацией;

- SLIP (Serial Line IP) IP для последовательных линий. Протокол последовательной посимвольной передачи данных;
- SNMP (Simple Network Management Protocol) простой протокол сетевого управления;
- FTAM (File Transfer, Access, and Management) протокол передачи, доступа и управления файлами.

Уровень представления данных (Presentation layer)

- Этот уровень обеспечивает то, что информация, передаваемая прикладным уровнем, будет понятна прикладному уровню в другой системе
- В случаях необходимости уровень представления в момент передачи информации выполняет преобразование форматов данных в некоторый общий формат представления, а в момент приема, соответственно, выполняет обратное преобразование. Таким образом, прикладные уровни могут преодолеть, например, синтаксические различия в представлении данных

- В основу общего представления данных положена единая для всех уровней модели система ASN.1. Эта система служит для описания структуры файлов, а также позволяет решить проблему шифрования данных. На этом уровне может выполняться шифрование и дешифрование данных, благодаря которым секретность обмена данными обеспечивается сразу для всех прикладных сервисов

- Примером такого протокола является протокол *Secure Socket Layer (SSL),* который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP. Этот уровень обеспечивает преобразование данных (кодирование, компрессия и т.п.) прикладного уровня в поток информации для транспортного уровня.

Представительный уровень выполняет следующие основные функции:

- Генерация запросов на установление сеансов взаимодействия прикладных процессов.
- Согласование представления данных между прикладными процессами.
- Реализация форм представления данных.
- Представление графического материала (чертежей, рисунков, схем).
- Засекречивание данных.
- Передача запросов на прекращение сеансов.

Сеансовый уровень (Session layer)

- Сеансовый уровень – это уровень, определяющий процедуру проведения сеансов между пользователями или прикладными процессами.

- Сеансовый уровень обеспечивает управление диалогом для того, чтобы фиксировать, какая из сторон является активной в настоящий момент, а также предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, вместо того чтобы начинать все сначала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется.

- Сеансовый уровень управляет передачей информации между прикладными процессами, координирует прием, передачу и выдачу одного сеанса связи. Кроме того, сеансовый уровень содержит дополнительно функции управления паролями, управления диалогом, синхронизации и отмены связи в сеансе передачи после сбоя вследствие ошибок в нижерасположенных уровнях. сообщений во время сеанса и завершение сеанса.

- Функции этого уровня состоят *в координации связи* между двумя прикладными программами, работающими на разных рабочих станциях. Это происходит в виде хорошо структурированного диалога. В число этих функций входит создание сеанса, управление передачей и приемом пакетов

- На сеансовом уровне определяется, какой будет передача между двумя прикладными процессами:
- *полудуплексной* (процессы будут передавать и принимать данные по очереди);
- *дуплексной* (процессы будут передавать данные, и принимать их одновременно).

- В полудуплексном режиме сеансовый уровень выдает тому процессу, который начинает передачу, *маркер данных*. Когда второму процессу приходит время отвечать, маркер данных передается ему. Сеансовый уровень разрешает передачу только той стороне, которая обладает маркером данных.

Сеансовый уровень обеспечивает выполнение следующих функций:

- Установление и завершение на сеансовом уровне соединения между взаимодействующими системами.
- Выполнение нормального и срочного обмена данными между прикладными процессами.
- Управление взаимодействием прикладных процессов.
- Синхронизация сеансовых соединений.
- Извещение прикладных процессов об исключительных ситуациях.

- Установление в прикладном процессе меток, позволяющих после отказа либо ошибки восстановить его выполнение от ближайшей метки.
- Прерывание в нужных случаях прикладного процесса и его корректное возобновление.
- Прекращение сеанса без потери данных.
- Передача особых сообщений о ходе проведения сеанса.

Транспортный уровень (Transport Layer)

- Транспортный уровень предназначен для передачи пакетов через коммуникационную сеть. На транспортном уровне пакеты разбиваются на блоки.

- Работа транспортного уровня заключается в том, чтобы обеспечить приложениям или верхним уровням модели (прикладному и сеансовому) передачу данных с той степенью надежности, которая им требуется.

классы сервиса, предоставляемые транспортным уровнем

- Эти виды сервиса отличаются качеством предоставляемых услуг:
- срочностью,
- возможностью восстановления прерванной связи,
- наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол,
- а главное способностью к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов.

- Транспортный уровень определяет адресацию физических устройств (систем, их частей) в сети. Этот уровень гарантирует доставку блоков информации адресатам и управляет этой доставкой. Его главной задачей является обеспечение эффективных, удобных и надежных форм передачи информации между системами. Когда в процессе обработки находится более одного пакета, транспортный уровень контролирует очередность прохождения пакетов. Если проходит дубликат принятого ранее сообщения, то данный уровень опознает это и игнорирует сообщение.

функции транспортного уровня

- Управление передачей по сети и обеспечение целостности блоков данных.
- Обнаружение ошибок, частичная их ликвидация и сообщение о неисправленных ошибках.
- Восстановление передачи после отказов и неисправностей.
- Укрупнение или разделение блоков данных.
- Предоставление приоритетов при передаче блоков (нормальная или срочная).
- Подтверждение передачи.
- Ликвидация блоков при тупиковых ситуациях в сети.

- Начиная с транспортного уровня, все вышележащие протоколы реализуются программными средствами, обычно включаемыми в состав сетевой операционной системы.
- Наиболее распространенные протоколы транспортного уровня включают в себя:
- TCP (Transmission Control Protocol) протокол управления передачей стека TCP/IP;
- UDP (User Datagram Protocol) пользовательский протокол дейтаграмм стека TCP/IP;

- NCP (NetWare Core Protocol) базовый протокол сетей NetWare;
- SPX (Sequenced Packet eXchange) упорядоченный обмен пакетами стека Novell;
- TP4 (Transmission Protocol) – протокол передачи класса 4.

Сетевой уровень (Network Layer)*******

- Сетевой уровень обеспечивает прокладку каналов, соединяющих абонентские и административные системы через коммуникационную сеть, выбор маршрута наиболее быстрого и надежного пути.

- Сетевой уровень устанавливает связь в вычислительной сети между двумя системами и обеспечивает прокладку виртуальных каналов между ними. *Виртуальный или логический канал* - это такое функционирование компонентов сети, которое создает взаимодействующим компонентам иллюзию прокладки между ними нужного тракта. Кроме этого, сетевой уровень сообщает транспортному уровню о появляющихся ошибках. Сообщения сетевого уровня принято называть *пакетами* (packet). В них помещаются фрагменты данных. Сетевой уровень отвечает за их адресацию и доставку.

- Прокладка наилучшего пути для передачи данных называется *маршрутизацией*, и ее решение является главной задачей сетевого уровня. Эта проблема осложняется тем, что самый короткий путь не всегда самый лучший. Часто критерием при выборе маршрута является время передачи данных по этому маршруту; оно зависит от пропускной способности каналов связи и интенсивности трафика, которая может изменяться с течением времени. Некоторые алгоритмы маршрутизации пытаются приспособиться к изменению нагрузки, в то время как другие принимают решения на основе средних показателей за длительное время. Выбор маршрута может осуществляться и по другим критериям, например, надежности передачи.

- Протокол канального уровня обеспечивает доставку данных между любыми узлами только в сети с соответствующей *типовой топологией*. Это очень жесткое ограничение, которое не позволяет строить сети с развитой структурой, например, сети, объединяющие несколько сетей предприятия в единую сеть, или высоконадежные сети, в которых существуют избыточные связи между узлами.

- Таким образом, внутри сети доставка данных регулируется канальным уровнем, а вот доставкой данных между сетями занимается сетевой уровень. При организации доставки пакетов на сетевом уровне используется понятие *номер сети*. В этом случае *адрес* получателя состоит из *номера сети* и *номера компьютера* в этой сети.

- Сети соединяются между собой специальными устройствами, называемыми маршрутизаторами. *Маршрутизатор* это устройство, которое собирает информацию о топологии межсетевых соединений и на ее основании пересылает пакеты сетевого уровня в сеть назначения. Для того чтобы передать сообщение от отправителя, находящегося в одной сети, получателю, находящемуся в другой сети, нужно совершить некоторое количество транзитных передач (hops) между сетями, каждый раз, выбирая подходящий маршрут. Таким образом, маршрут представляет собой последовательность маршрутизаторов, по которым проходит пакет.

- Например, у меня трассировка маршрута к сайту yandex.ru выглядит приблизительно так:
- Трассировка маршрута к yandex.ru [213.180.204.11] с максимальным числом прыжков 30:
- 1 * * * Превышен интервал ожидания для запроса.
- 2 10 ms 10 ms 10 ms bla4703.mns.ru [80.70.239.5]
- 3 10 ms 16 ms 10 ms core-239–214.bla-bla.ru [80.70.239.254]
- 4 10 ms 10 ms 10 ms core-239–221.bla-bla.ru [80.70.239.241]
- 5 15 ms 16 ms 16 ms ix1-m10.yandex.net [193.232.246.93]
- 6 16 ms 15 ms 16 ms einstein-vlan501.yandex.net [87.250.243.125]
- 7 16 ms 15 ms 16 ms hummer-vlan2.yandex.net [87.250.228.136]
- 8 16 ms 10 ms 16 ms yandex.ru [213.180.204.11]
- Трассировка завершена.

- Командой **tracert** мы инициируем отправку пакетов данных тому получателю, который указали — это может быть адрес сайта, имя компьютера в сети или IP-адрес. При этом пакеты проходят через все промежуточные системы (обычно это специальные сетевые устройства — машрутизаторы) между нашим компьютером и получателем. Таким образом, мы устанавливаем маршрут до пункта назначения и, что гораздо более важно — определяем время отклика (значение в миллисекундах) каждого промежуточного узла.

- На тех участках маршрута, где время отклика минимально, передача осуществляется быстрее всего — это значит, что канал не перегружен и данные проходят практически без помех. Попробуйте, например, трассировку маршрута к самому себе: команда **tracert localhost** или равнозначная **tracert 127.0.0.1**. Там же, где время отклика больше некоторого стандартного значения, мы получаем результат «Превышен интервал ожидания для запроса», что равносильно потере пакетов данных.

- Таким образом можно установить, в каком месте цепочки находится проблема. Если пакеты не доходят до самого пункта назначения — значит, проблема в нем. Если цепочка обрывается на середине — проблема в каком-то из промежуточных маршрутизаторов. При этом с другого компьютера или по другому маршруту (если такой существует) наш отвалившийся сайт может быть доступен. Если пакеты не выходят за пределы сети нашего провайдера — стало быть, там и проблема. Ну, а чтобы узнать, в каких случаях нужно просто установить модемное соединение или вставить кабель в разъем сетевой карты, попробуйте его выдернуть (предварительно дочитав статью) и провести диагностику самостоятельно.

- Сетевой уровень отвечает за деление пользователей на группы и маршрутизацию пакетов на основе преобразования MAC-адресов в сетевые адреса. Сетевой уровень обеспечивает также прозрачную передачу пакетов на транспортный уровень.

- tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
- listening on eth0, link-type EN10MB (Ethernet), capture size 68 bytes
- 10:10:26.720123 IP 5.199.170.228.http > 192.168.160.230.50144: . 4066639563:4066
- 640923(1360) ack 3126638222 win 123
- 10:10:26.720884 IP 192.168.160.230.50144 > 5.199.170.228.http: . ack 4294942816 win 54400
- 10:10:26.724762 IP 5.199.170.228.http > 192.168.160.230.50144: . 1360:2720(1360)
- ack 1 win 123
- 10:10:26.725285 IP mail.p-a.by.33149 > 193.232.248.2.domain: 30967+[|domain]

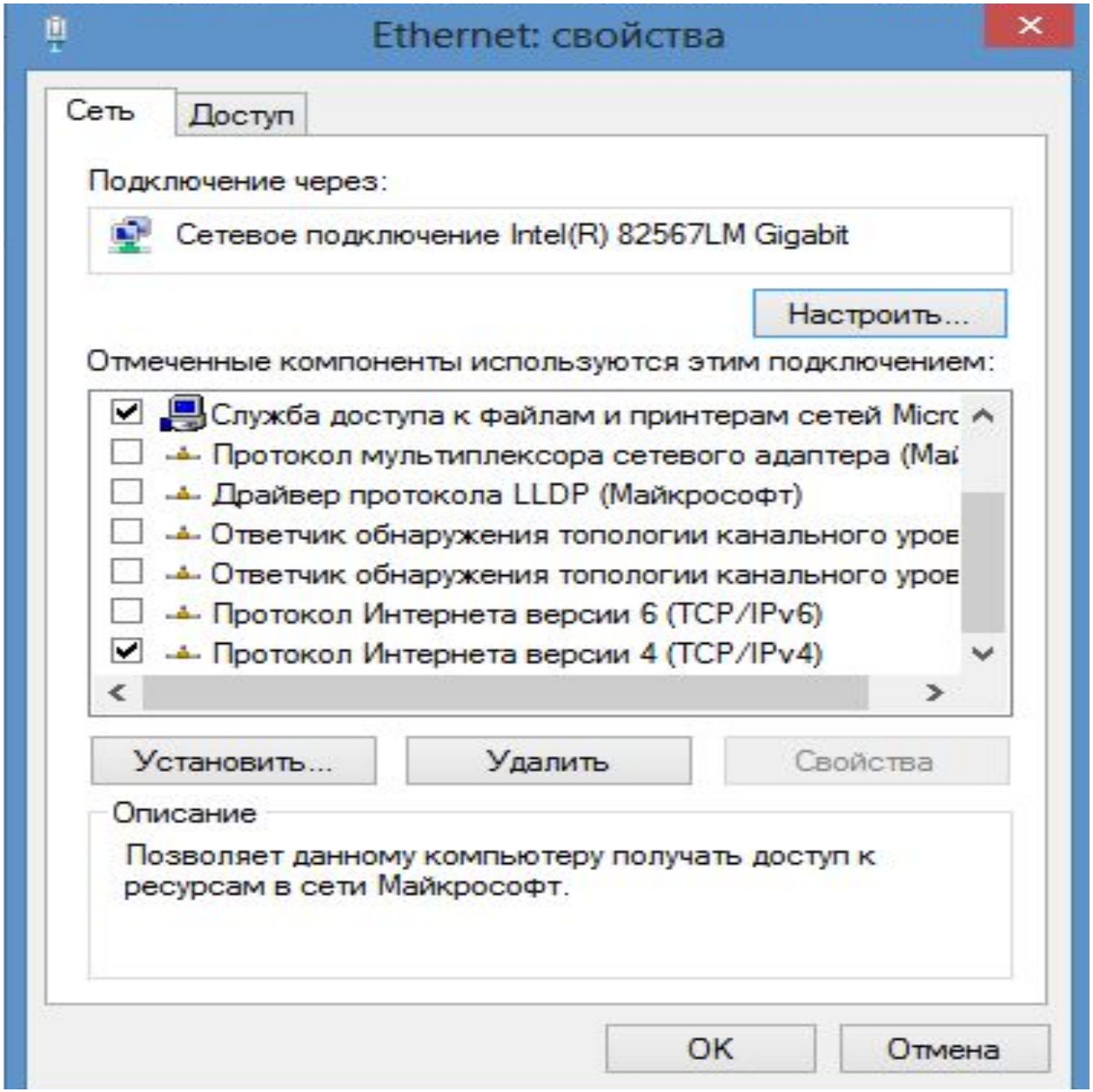
Сетевой уровень выполняет функции:

- Создание сетевых соединений и идентификация их портов.
- Обнаружение и исправление ошибок, возникающих при передаче через коммуникационную сеть.
- Управление потоками пакетов.
- Организация (упорядочение) последовательностей пакетов.
- Маршрутизация и коммутация.
- Сегментирование и объединение пакетов.

- На сетевом уровне определяется два вида протоколов. Первый вид относится к определению *правил передачи пакетов* с данными конечных узлов от узла к маршрутизатору и между маршрутизаторами. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня. Однако часто к сетевому уровню относят и другой вид протоколов, называемых *протоколами обмена маршрутной информацией*. С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений.

- Протоколы сетевого уровня реализуются программными модулями операционной системы, а также программными и аппаратными средствами маршрутизаторов.
- Наиболее часто на сетевом уровне используются протоколы:
- IP (Internet Protocol) протокол Internet, сетевой протокол стека TCP/IP, который предоставляет адресную и маршрутную информацию;
- IPX (Internetwork Packet Exchange) протокол межсетевого обмена пакетами, предназначенный для адресации и маршрутизации пакетов в сетях Novell;

- X.25 международный стандарт для глобальных коммуникаций с коммутацией пакетов (частично этот протокол реализован на уровне 2);
- CLNP (Connection Less Network Protocol) сетевой протокол без организации соединений.


Ethernet: свойства
Сеть
Доступ
Подключение через:
Сетевое подключение Intel(R) 82567LM Gigabit
Настроить...
Отмеченные компоненты используются этим подключением:
Служба доступа к файлам и принтерам сетей Micr
Протокол мультиплексора сетевого адаптера (Ma
Драйвер протокола LLDP (Майкрософт)
Ответчик обнаружения топологии канального уров
Ответчик обнаружения топологии канального уров
Протокол Интернета версии 6 (TCP/IPv6)
Протокол Интернета версии 4 (TCP/IPv4)
Установить...
Удалить
Свойства
Описание
Позволяет данному компьютеру получать доступ к ресурсам в сети Майкрософт.
OK
Отмена

Канальный уровень (Data Link)

- Единицей информации канального уровня являются *кадры (frame)*. Кадры – это логически организованная структура, в которую можно помещать данные. Задача канального уровня передавать кадры от сетевого уровня к физическому уровню.

- На физическом уровне просто пересылаются биты. При этом не учитывается, что в некоторых сетях, в которых линии связи используются попеременно несколькими парами взаимодействующих компьютеров, физическая среда передачи может быть занята. Поэтому одной из задач канального уровня является проверка доступности среды передачи. Другой задачей канального уровня является реализация механизмов обнаружения и коррекции ошибок.

- Канальный уровень обеспечивает корректность передачи каждого кадра, помещая специальную последовательность бит, в начало и конец каждого кадра, чтобы отметить его, а также вычисляет контрольную сумму, суммируя все байты кадра определенным способом и добавляя контрольную сумму к кадру. Когда кадр приходит, получатель снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой из кадра. Если они совпадают, кадр считается правильным и принимается. Если же контрольные суммы не совпадают, то фиксируется ошибка.

- Задача канального уровня - брать пакеты, поступающие с сетевого уровня и готовить их к передаче, укладывая в кадр соответствующего размера. Этот уровень обязан определить, где начинается и где заканчивается блок, а также обнаруживать ошибки передачи.

- На этом же уровне определяются правила использования физического уровня узлами сети. Электрическое представление данных в ЛВС (биты данных, методы кодирования данных и маркеры) распознаются на этом и только на этом уровне. Здесь обнаруживаются и исправляются (путем требований повторной передачи данных) ошибки.

- Канальный уровень обеспечивает создание, передачу и прием кадров данных. Этот уровень обслуживает запросы сетевого уровня и использует сервис физического уровня для приема и передачи пакетов. Спецификации IEEE 802.X делят канальный уровень на два подуровня:

- *LLC* (*Logical Link Control*) управление логическим каналом осуществляет логический контроль связи. Подуровень LLC обеспечивает обслуживание сетевого уровня и связан с передачей и приемом пользовательских сообщений.

- *MAC* (*Media Assess Control)* контроль доступа к среде. Подуровень MAC регулирует доступ к разделяемой физической среде (передача маркера или обнаружение коллизий или столкновений) и управляет доступом к каналу связи. Подуровень *LLC* находится выше подуровня *MAC*.

- Канальный уровень определяет доступ к среде и управление передачей посредством процедуры передачи данных по каналу. При больших размерах передаваемых блоков данных канальный уровень делит их на кадры и передает кадры в виде последовательностей. При получении кадров уровень формирует из них переданные блоки данных. Размер блока данных зависит от способа передачи, качества канала, по которому он передается.

- В локальных сетях протоколы канального уровня используются компьютерами, мостами, коммутаторами и маршрутизаторами. В компьютерах функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов.

Канальный уровень может выполнять следующие виды функций:

- Организация (установление, управление, расторжение) канальных соединений и идентификация их портов.
- Организация и передача кадров.
- Обнаружение и исправление ошибок.
- Управление потоками данных.
- Обеспечение прозрачности логических каналов (передачи по ним данных, закодированных любым способом).

Наиболее часто используемые протоколы на канальном уровне включают:

- HDLC (High Level Data Link Control) протокол управления каналом передачи данных высокого уровня, для последовательных соединений;
- IEEE 802.2 LLC (тип I и тип II) обеспечивают MAC для сред 802.x;
- Ethernet сетевая технология по стандарту IEEE 802.3 для сетей, использующая шинную топологию и коллективный доступ с прослушиванием несущей и обнаружением конфликтов;

- Token ring сетевая технология по стандарту IEEE 802.5, использующая кольцевую топологию и метод доступа к кольцу с передачей маркера;
- FDDI (Fiber Distributed Date Interface Station) сетевая технология по стандарту IEEE 802.6, использующая оптоволоконный носитель;
- X.25 международный стандарт для глобальных коммуникаций с коммутацией пакетов;
- Frame relay сеть, организованная из технологий X25 и ISDN.

Физический уровень (Physical Layer)

- Физический уровень предназначен для сопряжения с *физическими средствами соединения*. *Физические средства соединения* – это совокупность *физической среды*, аппаратных и программных средств, обеспечивающая передачу сигналов между системами. *Физическая среда* – это материальная субстанция, через которую осуществляется передача сигналов. Физическая среда является основой, на которой строятся физические средства соединения. В качестве физической среды широко используются эфир, металлы, оптическое стекло и кварц.

Физический уровень состоит из *Подуровня стыковки со средой* и *Подуровня преобразования передачи*.

- Первый из них обеспечивает сопряжение потока данных с используемым физическим каналом связи.
- Второй осуществляет преобразования, связанные с применяемыми протоколами.

- Физический уровень обеспечивает физический интерфейс с каналом передачи данных, а также описывает процедуры передачи сигналов в канал и получения их из канала. На этом уровне определяются электрические, механические, функциональные и процедурные параметры для физической связи в системах.

- Физический уровень получает пакеты данных от вышележащего канального уровня и преобразует их в оптические или электрические сигналы, соответствующие 0 и 1 бинарного потока. Эти сигналы посылаются через среду передачи на приемный узел.

- Механические и электрические / оптические свойства среды передачи определяются на физическом уровне и включают:
- тип кабелей и разъемов;
- разводку контактов в разъемах;
- схему кодирования сигналов для значений 0 и 1.

Физический уровень выполняет следующие функции: ********

- Установление и разъединение физических соединений.
- Передача сигналов в последовательном коде и прием.
- Прослушивание, в нужных случаях, каналов.
- Идентификация каналов.
- Оповещение о появлении неисправностей и отказов.

- Оповещение о появлении неисправностей и отказов связано с тем, что на физическом уровне происходит обнаружение определенного класса событий, мешающих нормальной работе сети (столкновение кадров, посланных сразу несколькими системами, обрыв канала, отключение питания, потеря механического контакта и т. д.).

- Виды сервиса, предоставляемого канальному уровню, определяются протоколами физического уровня. Прослушивание канала необходимо в тех случаях, когда к одному каналу подключается группа систем, но одновременно передавать сигналы разрешается только одной из них.

- Поэтому прослушивание канала позволяет определить, свободен ли он для передачи. В ряде случаев для более четкого определения структуры физический уровень разбивается на несколько подуровней. Например, физический уровень беспроводной сети делится на три подуровня

1c	Подуровень, не зависимый от физических средств соединения
1б	Переходный подуровень,
1а	Подуровень, зависимый от физических средств соединения

- Функции физического уровня реализуются во всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером. Повторители являются единственным типом оборудования, которое работает только на физическом уровне

- Выполняется преобразование данных, поступающих от более высокого уровня, в сигналы передающие по кабелю. В глобальных сетях на этом уровне могут использоваться модемы и интерфейс *RS-232C*. В локальных сетях для преобразования данных применяют сетевые адаптеры, обеспечивающие скоростную передачу данных в цифровой форме. Пример протокола физического уровня - это широко известный интерфейс *RS-232C / CCITT V.2*, который является наиболее широко распространенной стандартной последовательной связью между компьютерами и периферийными устройствами.

- Можно считать этот уровень, отвечающим за аппаратное обеспечение.
- Физический уровень может обеспечивать как асинхронную (последовательную) так и синхронную (параллельную) передачу, которая применяется для некоторых мэйнфреймов и мини - компьютеров. На Физическом уровне должна быть определена схема кодирования для представления двоичных значений с целью их передачи по каналу связи. Во многих локальных сетях используется манчестерское кодирование.

- Примером протокола физического уровня может служить спецификация 10Base-T технологии Ethernet, которая определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с волновым сопротивлением 100 Ом, разъем RJ-45, максимальную длину физического сегмента 100 метров, манчестерский код для представления данных на кабеле, и другие характеристики среды и электрических сигналов.

- В манчестерском коде для кодирования единиц и нулей используется перепад потенциала, то есть фронт импульса. При манчестерском кодировании каждый такт делится на две части. Информация кодируется перепадами потенциала, происходящими в середине каждого такта. Единица кодируется перепадом от низкого уровня сигнала к высокому, а ноль - обратным перепадом. В начале каждого такта может происходить служебный перепад сигнала, если нужно представить несколько единиц или нулей подряд.

- Так как сигнал изменяется по крайней мере один раз за такт передачи одного бита данных, то манчестерский код обладает хорошими самосинхронизирующими свойствами. Полоса пропускания манчестерского кода уже, чем у биполярного импульсного. У него также нет постоянной составляющей, а основная гармоника в худшем случае (при передаче последовательности единиц или нулей) имеет частоту N Гц, а в лучшем (при передаче чередующихся единиц и нулей) она равна N/2 Гц, как и у кодов AMI или NRZ.

- В среднем ширина полосы манчестерского кода в полтора раза уже, чем у биполярного импульсного кода, а основная гармоника колеблется вблизи значения 3N/4. Манчестерский код имеет еще одно преимущество перед биполярным импульсным кодом. В последнем для передачи данных используются три уровня сигнала, а в манчестерском - два.

К числу наиболее распространенных спецификаций физического уровня относятся:

- EIA-RS-232-C, CCITT V.24/V.28 - механические/электрические характеристики несбалансированного последовательного интерфейса;
- EIA-RS-422/449, CCITT V.10 - механические, электрические и оптические характеристики сбалансированного последовательного интерфейса;

- Ethernet – сетевая технология по стандарту IEEE 802.3 для сетей, использующая шинную топологию и коллективный доступ с прослушиванием несущей и обнаружением конфликтов;
- Token ring – сетевая технология по стандарту IEEE 802.5, использующая кольцевую топологию и метод доступа к кольцу с передачей маркера;

Сетезависимые протоколы

- Функции всех уровней модели OSI могут быть отнесены к одной из двух групп: либо к функциям, зависящим от конкретной технической реализации сети, либо к функциям, ориентированным на работу с приложениями.
- Три нижних уровня физический, канальный и сетевой являются сетезависимыми, протоколы этих уровней тесно связаны с технической реализацией сети, с используемым коммуникационным оборудованием. Например, переход на оборудование FDDI означает смену протоколов физического и канального уровня во всех узлах сети.

- Три верхних уровня сеансовый, уровень представления и прикладной ориентированы на приложения и мало зависят от технических особенностей построения сети. На протоколы этих уровней не влияют никакие изменения в топологии сети, замена оборудования или переход на другую сетевую технологию. Так, переход от Ethernet на высокоскоростную технологию 100VG-AnyLAN не потребует никаких изменений в программных средствах, реализующих функции прикладного, представительного и сеансового уровней.

- Транспортный уровень является промежуточным, он скрывает все детали функционирования нижних уровней от верхних уровней. Это позволяет разрабатывать приложения, не зависящие от технических средств, непосредственно занимающихся транспортировкой сообщений.
- Одна рабочая станция взаимодействует с другой рабочей станцией посредством протоколов всех семи уровней. Это взаимодействие станции осуществляют через различные коммуникационные устройства: концентраторы, модемы, мосты, коммутаторы, маршрутизаторы, мультиплексоры

- В зависимости от типа коммуникационное устройство может работать:
- либо только на физическом уровне (повторитель);
- либо на физическом и канальном уровнях (мост);
- либо на физическом, канальном и сетевом уровнях, иногда захватывая и транспортный уровень (маршрутизатор).

- Модель OSI представляет собой хотя и очень важную, но только одну из многих моделей коммуникаций. Эти модели и связанные с ними стеки протоколов могут отличаться количеством уровней, их функциями, форматами сообщений, сервисами, предоставляемыми на верхних уровнях, и прочими параметрами.

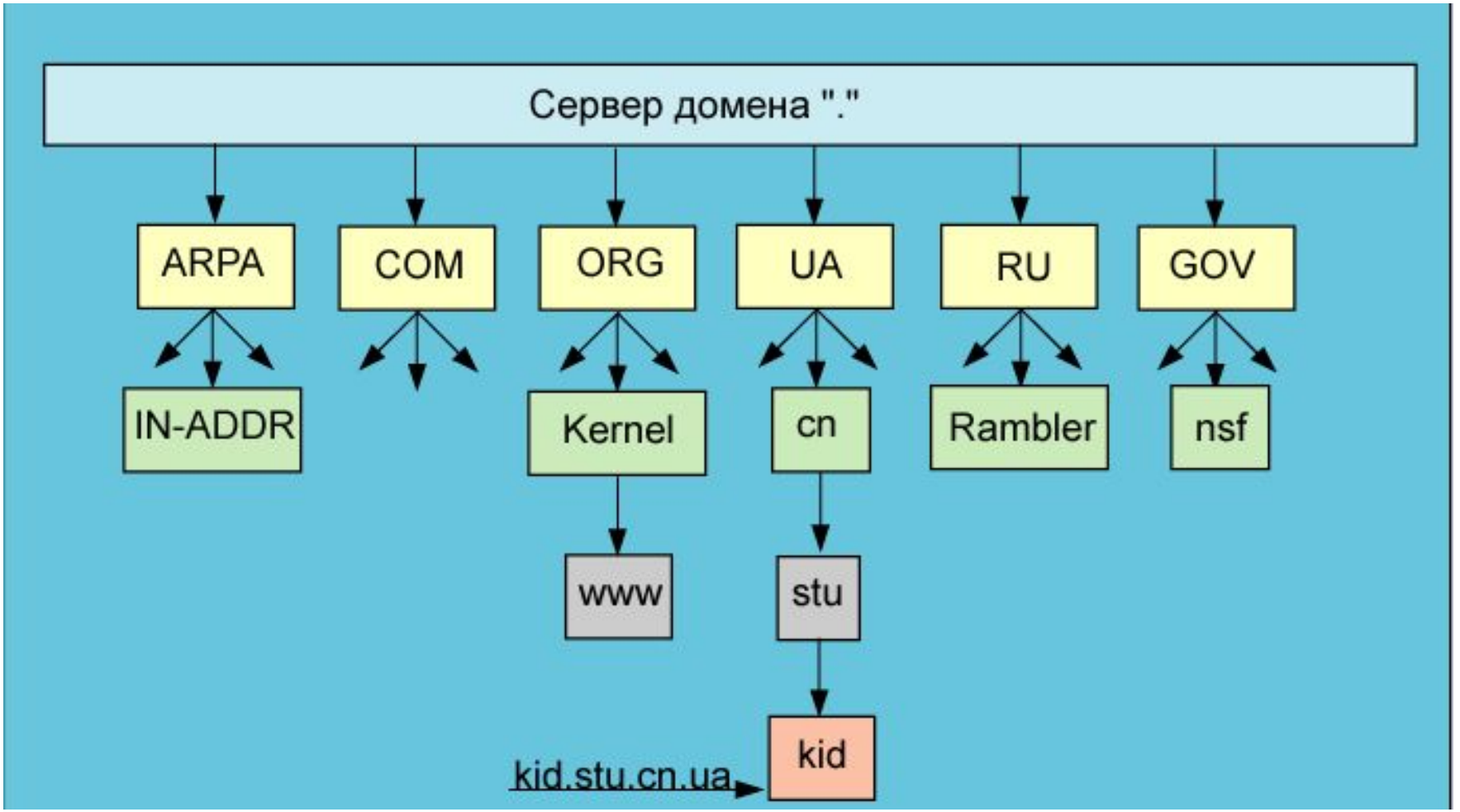
Стеки коммуникационных протоколов

- Иерархически организованная совокупность протоколов, решающих задачу взаимодействия узлов сети, называется *стеком коммуникационных протоколов*.
- Протоколы соседних уровней, находящихся в одном узле, взаимодействуют друг с другом также в соответствии с четко определенными правилами и с помощью стандартизованных форматов сообщений. Эти правила принято называть *интерфейсом*. Интерфейс определяет набор услуг, которые нижележащий уровень предоставляет вышележащему уровню.

Служба имен в Интернете DNS

- Служба доменных имён DNS является важнейшей системной службой в TCP/IP сетях. Назначение DNS – преобразование символьных имен в IP адреса и наоборот, а так же предоставление дополнительной информации о хостах и группах хостов (доменах), такой как адреса почтовых обменников, публичные ключи служб и т.п.

- В сети Интернет служба DNS оперирует распределённой иерархической базой данных в виде дерева имен, находящейся на множестве различных хостов. Ответственность за корректность базы данных в каждом узле дерева возлагается на администратора соответствующего домена.


Сервер домена "."
ARPA
COM
ORG
UA
RU
GOV
IN-ADDR
Kernel
cn
Rambler
nsf
www
stu
kid
kid.stu.cn.ua

- В сети Интернет корнем дерева является домен “.”. Полное - абсолютное или полностью определенное, fully qualified domain name - доменное имя заканчивается точкой, обозначающей корень доменного дерева, но часто эта завершающая точка опускается. Анализ имени производится справа налево. Самая правая секция имени характеризует страну (для каждой страны мира выделен свой домен с двух символьным именем в соответствии со стандартом ISO, например, ua – Украина, ru – Россия, uk – Англия и т.п.) или характер организации (образовательная, коммерческая, правительственная и т.п.)

- Таким образом, в службе DNS каждый сервер отвечает за определенную зону (зона ответственности) - т.е. свою часть дерева доменных имен, хранит соответствующие базы данных и отвечает на запросы. При этом вышестоящие по дереву серверы имеют информацию об адресах нижестоящих серверов, что обеспечивает связность дерева. Говорят, что вышестоящий сервер делегирует нижестоящему серверу полномочия по обслуживанию определенной зоны.
- Важно понимать различие между доменом и зоной. Домен - это поддерево дерева доменных имен. Зона - это часть дерева, за которую отвечает тот или иной DNS-сервер

- За каждую зону DNS отвечает не менее двух серверов. Один из них является первичным, primary, или, в новой терминологии — master. остальные - вторичными, secondary, или slave. Первичный сервер содержит оригинальные файлы с базой данных DNS для своей зоны. Вторичные серверы получают эти данные по сети от первичного сервера и периодически запрашивают первичный сервер на предмет обновления данных.

- Признаком обновления данных служит увеличение серийного номера в записи SOA – см ниже. В случае, если данные на первичном сервере обновлены, вторичный сервер запрашивает "передачу зоны" ("zone transfer")- т.е. базы данных требуемой зоны. Передача зоны происходит с помощью протокола TCP, порт 53, в отличие от запросов, которые направляются на UDP/53

- Изменения в базу данных DNS могут быть внесены только на первичном сервере. С точки зрения обслуживания клиентских запросов первичные и вторичные серверы идентичны. Рекомендуется, чтобы первичный и вторичные серверы находились в разных сетях - для увеличения надежности обработки запросов на случай, если сеть одного из серверов становится недоступной. Серверы DNS не обязаны находиться в том домене, за который они отвечают.

- Вторичный сервер необязательно получает данные непосредственно с первичного сервера; источником данных может служить и другой вторичный сервер. В любом случае сервер-источник данных для данного вторичного сервера называется "главным" ("master").

- Процесс получения собственного домена называется “делегирование”, что, собственно, и отражает суть действий: субдомен передается в полное управление его администратору. Для получения субдомена в каком-то домене необходимо в соответствии с полиси обратится к регистратору, выполнить формальные процедуры, а так же ряд технических действий по настройке сервера DNS

- Различают 3 режима работы сервера DNS:
- ***master*** (primary). Данный режим используется администратором зоны, файлы баз данных ведутся вручную на этом сервере. Данный сервер является абсолютным авторитетным источником информации для данной зоны;
- ***slave*** (secondary). Данный режим используется по просьбе администратора зоны, которая автоматически регулярно копируется с master сервера. Данный сервер является авторитетным источником информации для данной зоны;

- ***hint*** (caching). Режим кэширования всех запросов, попадающих в определённую зону, обычно “.”, т.е. кэшируются все запросы. Такой сервер обычно используется для ускорения работы с сетью.

- Для каждой зоны, обслуживаемой данным сервером, может быть выбран тот или иной режим. Обычно для зоны “.” все сервера конфигурируются по типу hint, что позволяет кешировать все запросы пользовательских рабочих станций на время жизни конкретной записи DNS. Это значительно ускоряет обработку локальных запросов.

- **Записи ресурсов в базе данных домена**
- Файл любой зоны начинается с записи Start Of Authority, ***SOA***. Эта запись является заголовочной и содержит информацию о размещении зоны, о почтовом адресе ответственного лица и о базовых временных параметрах записей данной зоны.
- Файл прямой зоны содержит стандартные записи ресурсов базы данных DNS для преобразования доменных имен хостов в данной зоне в IP-адреса, определения авторитарных DNS-серверов данной зоны, определения хостов-обработчиков почты для доменных имен в данной зоне и др.

- Файлы баз данных DNS состоят из стандартных записей ресурсов. В общем виде стандартная запись ресурса связывает данные определенного типа с некоторым именем и формируется по шаблону:
- ***имя [время_жизни_записи] IN тип_записи данные***
- Именем является некоторое доменное имя (необязательно имя физически существующих хоста или домена). Если поле "имя" пусто, то значение этого поля берется из предыдущей записи. Данными может быть, например, IP-адрес хоста, если имя относится к хосту, или DNS-сервер домена, если имя относится к домену, и т.п.

- Время жизни записи определяет время хранения информации этой записи в кэше запросившего запись сервера в секундах и указывается, только если оно отличается от времени жизни, определенного для всей зоны в записи SOA.
- Основные типы записей:
- ***SOA*** (Start Of Authority) - заголовок зоны;
- ***NS*** (Name Server) - сервер DNS;
- ***A*** (Address) - IP-адрес для хоста;
- ***MX*** (Mail Exchanger) - почтовый обменник;
- ***CNAME*** (Canonical Name) - каноническое имя, псевдоним хоста;
- ***PTR*** (Pointer) - указатель по обратной зоне, фактически — имя хоста;

- Рассмотрим примеры файлов базы данных DNS. Первой рассмотрим прямую зону для приватной части корпоративной сети, домен “stu.”, файл db.stu.

- Начасть с этого 1111111111111111

```
$ORIGIN .
stu 28800 IN SOA ns.stu. dnsmaster.stu. (
                2005033100 ; Serial
                28800 ; Refresh
                7200 ; Retry
                604800 ; Expire
                86400 ; Time To Live)
; authoritative name servers for zone
28800 IN NS ns.stu.
28800 IN NS ns1.stu.
```

- ; mail exchangers for entire zone
  28800 IN MX 10 stalker.stu.
  28800 IN MX 20 cs.stu.
- $ORIGIN stu.
  ; name servers glue records
  ns IN A 192.168.0.10
  ns1 IN A 192.168.0.14
  ;servers
  dragon IN A 192.168.0.17
  auth IN CNAME dragon.stu.
  cs IN A 192.168.0.14
  stalker IN A 192.168.0.10

- www IN CNAME stalker.stu.
  mail IN CNAME stalker.stu.
  ftp IN CNAME stalker.stu.
  www.docs IN CNAME cs.stu.
  kid IN A 192.168.0.12
  ; workstations
  ie-21-7 IN A 192.168.3.40
  ie-21-8 IN A 192.168.3.41
  ie-21-9 IN A 192.168.3.42
  vc-105-1 IN A 192.168.66.2

- Первая строка – это макрос, говорящий, что все имена далее следуют непосредственно за доменом “точка”. Таким образом, для приватной сети мы используем имена в нашем приватном дереве относительно нашего собственного корня “.”. Следует помнить, что для сервера, разрешающего одновременно и имена в корпоративной сети, и имена в интернете, имя зоны следует выбирать из 3-х символов, не совпадающих с именами TLD.

- Первой записью всегда идет ***SOA*** (Start of Authority), в которой указывается имя зоны (“stu.”, или макрос @), TTL, т.е. время жизни этой записи, дальше – ключевые слова IN (Internet records) и ***SOA***. Далее идут параметры зоны: имя основного сервера DNS, почтовый адрес администратора зоны, однако вместо символа “@” там стоит точка, поскольку @ - это ссылка на имя зоны.

- Сразу за открывающейся скобкой находится серийный номер данного файла, обычно в формате ггггммддNN. Серийный номер необходимо увеличивать при каждом изменении файла, что бы ведомые сервера идентифицировали изменения и обновили файлы баз данных с главного сервера. Далее следуют стандартные времена в секундах для данной зоны:

- Refresh – время, по истечении которого вторичные сервера должны обновить данные с первичных серверов (zone transfer);
- Retry – время, через которое вторичные сервера должны совершить повторную попытку обновления, если предыдущая попытка не удалась;
- Expire – время, через которое вторичные сервера должны выбросить запись о зоне и считать ее недоступной, если обновления не удались.
- TTL – стандартное время жизни записей из данной зоны для кеширующих серверов.

- Следующая группа записей является так же обязательной и указывает на авторитетные сервера имен для данной зоны - записи типа ***NS***. Авторитетным является сервер, на котором информация соответствует реальному состоянию зоны, т.е. регулярно обновляется (см. выше). Крайне желательно, чтобы имена, указанные в этой секции, имели соответствующие адресные ***IN A*** записи в этой же базе данных.

- Ниже следует секция почтовых обменников, т.е. записи типа ***MX*** (Mail Exchanger). Они указывают на сервера электронной почты, способные принимать почту для всего домена по протоколу SMTP. Чем меньше цифра перед именем, тем больший приоритет имеет данный почтовый сервер. Как правило, запись с наивысшим приоритетом относится к серверу, на котором почта заканчивает свой путь, а другие записи относятся к серверам-релеям, на которых почта может сохраняться некоторое время, пока основной почтовый сервер для зоны не доступен.

- Естественно, записи ***MX*** на релеи нельзя расставлять произвольно, поскольку релей обязательно должен быть сконфигурирован для приёма почты данного домена. При отсутствии записи MX для какого-либо доменного имени, почта, адресованная с этим доменным именем, будет доставляться непосредственно на хост, имеющий такое имя. Однако, такого хоста может не быть, в этом случае почта вернется отправителю с сообщением об ошибке.

- Ниже, после макроса “$ORIGIN stu.”, задающего суффикс для всех записей ниже, следуют записи типа ***IN A***, предназначенные для задания соответствия между именем хоста в зоне и его IP адресом.
- Для задания псевдонимов хостам используется запись ***CNAME*** (Canonical Name). Псевдонимы удобны для указания на стандартные сервисы, такие как www, mail, ftp, а так же для задания псевдонимов, используемых для создания виртуальных серверов (см. www, docs).

- Рассмотрим теперь файл зоны stu.cn.ua. Данная зона мало чем отличается от предыдущей зоны внешне.
- $ORIGIN .
  stu.cn.ua 28800 IN SOA ns.stu.cn.ua.
  nsmaster.stu.cn.ua(
  2005033100 ; Serial
  28800 ; Refresh
  7200 ; Retry
  604800 ; Expire
  86400 ; Time To Live
)
- ; authoritative name servers for zone
  IN NS ns.stu.cn.ua.
  IN NS ns1.stu.cn.ua.
  IN NS ns.cn.ua.

- ; mail exchangers for entire zone
  IN MX 10 stalker.stu.cn.ua.
  IN MX 15 cs.stu.cn.ua.
  IN MX 20 relay1.cn.ua
- ; name servers glue records
  ns.cn.ua IN A 212.86.96.10
  $ORIGIN stu.cn.ua.
  ns IN A 195.69.76.130
  ns1 IN A 195.69.76.134
  ;servers
  dragon IN A 195.69.76.137
  auth IN CNAME dragon.stu.cn.ua.
  cs IN A 195.69.76.134
  stalker IN A 195.69.76.130
  www IN CNAME stalker.stu.cn.ua.
  mail IN CNAME stalker.stu.cn.ua.
  ftp IN CNAME stalker.stu.cn.ua.
  www.docs IN CNAME cs.stu.cn.ua.
  ; workstations
  admin IN A 195.69.76.139

- Конфигурация данной зоны практически повторяет предыдущую зону, однако отличие в том, что данная зона является субдоменом домена cn.ua. Значит, она должна быть делегирована в соответствующей зоне cn.ua примерно так, как показано в следующем фрагменте:
-
- $ORIGIN cn.ua.
  stu IN NS ns.stu.cn.ua.
  IN NS ns1.stu.cn.ua.
  IN NS ns.cn.ua.
- $ORIGIN .
  ns.stu.cn.ua IN A 195.69.76.130
  ns1.stu.cn.ua IN A 195.69.76.134
-

- Как видно из приведенного фрагмента, в “материнской” зоне cn.ua. находятся только записи о серверах имен для делегированной зоны stu.cn.ua. Управление остальной информационной базой зону передается на сервера ns.stu.cn.ua и ns1.stu.cn.ua.

- **Обратная зона DNS начать**
- Теперь рассмотрим файлы обратных зон, предназначенные для проведения обратного DNS-преобразования, т.е. "IP-адрес в доменное имя".
- Для приватных блоков адресов, таких как 10.0.0.0/8, 172.16.0.0/12 и 192.168.0.0/16 никаких проблем с делегированием, в общем-то, нет, поскольку эти адреса не маршрутизируются в Интернете и нужны только внутри приватной сети.

- $ORIGIN .
- 0.168.192.IN-ADDR.ARPA. 86400 IN SOA ns.stu. dnsmaster.stu. (
  2006060200
  86400
  14400
  3600000
- 345600
-)
  86400 IN NS ns.stu.
  86400 IN NS ns1.stu.
- $ORIGIN 0.168.192.IN-ADDR.ARPA.
- 10 IN PTR stalker.stu.
  14 IN PTR cs.stu.
  17 IN PTR dragon.stu.
- 12 IN PTR kid.stu.

- Стоит обратить внимание, что имя зоны состоит из развёрнутых по отношению к записи адреса цифр. Для адресного блока 192.168.0.0/24 имя зоны 0.168.192.IN-ADDR.ARPA. INADDR.ARPA. - это специальный домен верхнего уровня, отведенный для делегирования обратных зон. В файле обратной зоны присутствует, конечно же, запись ***SOA***, как минимум пара записей типа ***NS*** об официальных авторитетных серверах и записи типа ***PTR*** (Pointer), ставящие в соответствие адреса и имена.

- **Обратная зона для публичных адресов**
- $ORIGIN .
- 76.69.195.IN-ADDR.ARPA. 86400 IN SOA ns.stu.cn.ua dnsmaster.stu.cn.ua (2004060200 86400 14400 3600000 345600)
- IN NS ns.stu.cn.ua.
  IN NS ns1.stu.cn.ua.
- $ORIGIN 0.168.192.IN-ADDR.ARPA.
- 10 IN PTR stalker.stu.cn.ua.
  14 IN PTR cs.stu.cn.ua.
  17 IN PTR dragon.stu.cn.ua.
- 12 IN PTR kid.stu.cn.ua.
-

- Отличие данной зоны от предыдущей опять же только в том, что она является публичной и должна делегироваться в соответствии с правилами выдачи и регистрации IP адресов

Организация службы электронной почты в Интернет

Электронная почта - это служба пересылки сообщений между зарегистрированными адресами

Адрес электронной почты состоит выглядит как

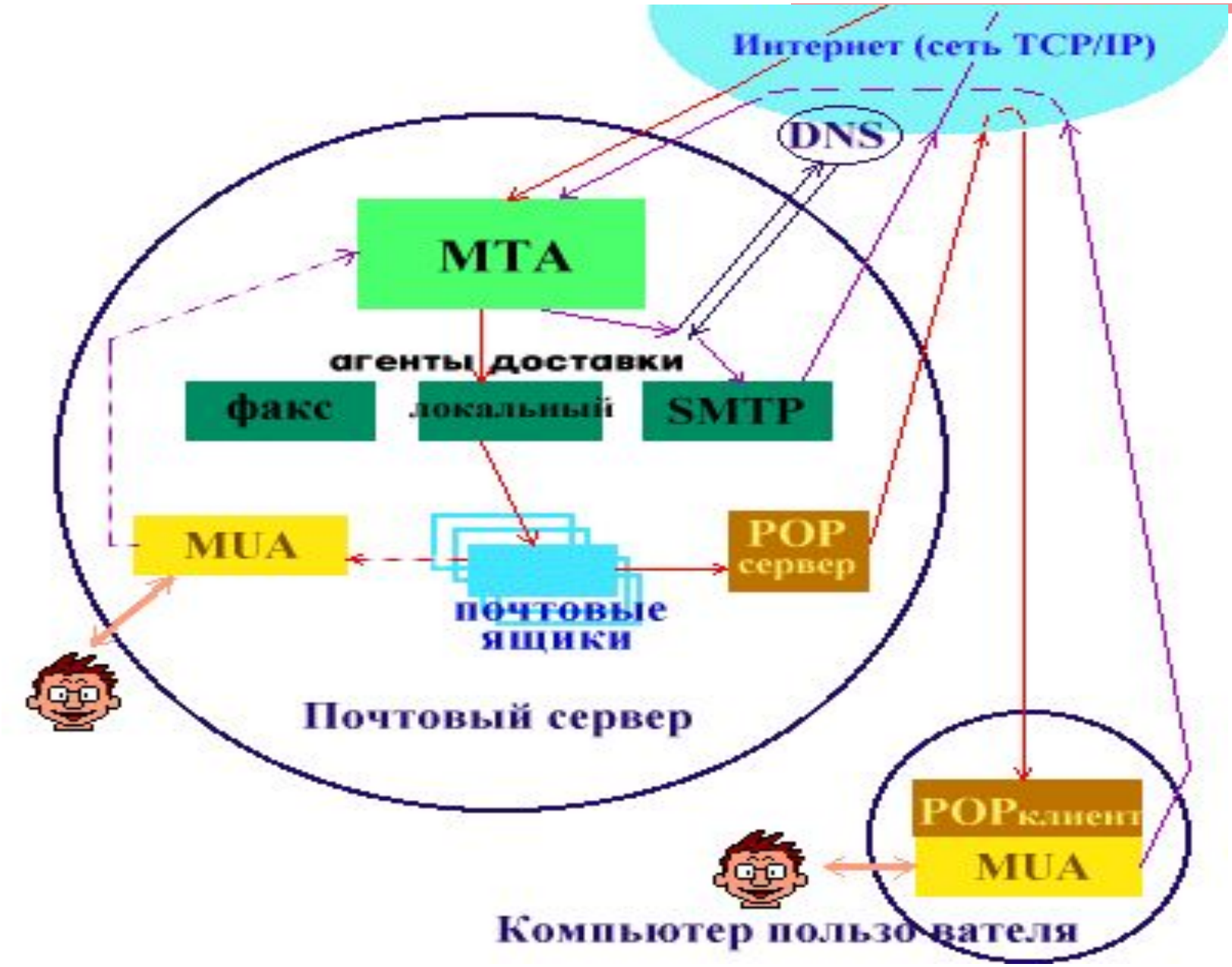
почтовый_ящик@почтовый.домен

m2@yandex.ru Yachik@gmail.com

- где *почтовый.домен* - некое доменное имя, а *почтовый_ящик* - имя-идентификатор корреспондента. Почтовый ящик может соответствовать одному человеку, группе людей, официальному почтовому адресу, автомату-обработчику и т.д. - форма адреса от этого не зависит. Далее в этом пункте будем считать, что почтовые ящики являются персональными. Почтовый домен - это доменное имя, для которого в системе DNS существует запись типа MX (см. тему "DNS"), либо запись типа A, если MX отсутствует

- Компьютер, на который указывает запись MX, является *почтовым сервером* для данного почтового домена. Вся почта, направленная на адреса в данном почтовом домене, поступает на этот сервер, которые принимает решение о том, как дальше поступить с этой почтой

- Основную роль в системе электронной почты играют программы трех типов:
- транспортные агенты (MTA - Mail Transport Agent),
- агенты доставки (MDA - Mail Delivery Agent),
- пользовательские агенты (MUA - Mail User Agent).


Интернет (сеть TCP/IP)
DNS
MTA
агенты доставки
факс
локальный
SMTP
MUA
POP
сервер
почтовые
ящики
Почтовый сервер
POPклиент
MUA
Компьютер пользователя

- Транспортный агент работает, как правило, на почтовом сервере. Транспортный агент функционирует как маршрутизатор почтовых сообщений. Его функции следующие:
- анализ и преобразование адресов и заголовков почтовых сообщений, в том числе:
  - разбор списков рассылки, пседонимов, переадресации (форвардинг),

- преобразование адресов в формат другой почтовой системы, если MTA функционирует как шлюз между двумя почтовыми системами (например, между Internet Mail и Sprint Mail),
- преобразование имени почтового домена отправителя (маскарад),
- установка служебных заголовков в сообщении, отражающих его маршрут и процесс обработки;

- опрос DNS на предмет имени и адреса почтового сервера адресата сообщения;
- определение агента доставки для каждого сообщения и передача сообщения выбранному агенту доставки;
- управление очередью сообщений, отложенный и повторный вызов агентов доставки в случае невозможности немедленной доставки сообщения;
- возврат сообщений, которые по каким-либо причинам невозможно доставить по назначению.

- Агент доставки производит доставку сообщения каким-либо специфическим способом. Существует несколько стандартных типов агентов доставки:
- local - письмо направлено на почтовый ящик, находящийся на этом же компьютере; доставка производится, например, добавлением содержимого сообщения в определенный файл (в Unix это файл /var/mail/*почтовый_ящик*).

- SMTP - письмо направлено на почтовый ящик в другом почтовом домене; доставка производится путем соединения с транспортным агентом на удаленном сервере с помощью протокола SMTP.
- prog - письмо должно быть обработано какой-либо программой; доставка производится вызовом этой программы, на вход которой подается содержимое письма.

- Вообще методы доставки (и, соответственно, агенты) могут быть разнообразными: например, сохранение письма в базе данных; пересылка письма по факсу и т.д. Выбор агента доставки для каждого конкретного письма производится транспортным агентом в соответствии с заданной конфигурацией транспортного агента и адресом назначения письма.

- Пользовательский агент является оболочкой пользователя для работы с электронной почтой, его функции:
- получение сообщений с почтового сервера;
- презентация, хранение, удаление и каталогизирование почтовых сообщений;
- создание нового сообщения и передача его транспортному агенту для дальнейшей обработки и доставки.

- Рассмотрим входящее сообщение (красные стрелки) от bg@aquarium.ru к ivanov@cts.vvsu.ru. Сообщение поступает по сети к транспортному агенту. (Для передачи сообщений транспортному агенту по сети используется протокол SMTP, рассмотренный в соответствующем пункте ниже.) MTA, проанализировав заголовок сообщения, определяет, что оно адресовано в почтовый домен cts.vvsu.ru, который он обслуживает. В соответствии с этим выбирается агент доставки local, запускается программа этого агента и на

- вход ей подается текст сообщения со всеми заголовками. Агент доставки каким-то способом, не интересным для транспортного агента, производит доставку сообщения и прекращает свою работу. Транспортный агент анализирует статус выхода (exit status) программы агента доставки, по которому определяет было ли сообщение успешно доставлено или произошла ошибка.

- В случае ошибки MTA формирует сообщение об ошибке, исходящее с адреса MAILER-DAEMON@m.vvsu.ru, которое будет отправлено отправителю письма (и, как правило, администратору почтового сервера по адресу postmaster@m.vvsu.ru). В случае успешного завершения работы агента доставки письмо считается доставленным получателю.

- Агент доставки local (в Unix это программа mail, запущенная как "mail -d ivanov") производит доставку методом добавления содержимого письма к файлу /var/mail/ivanov (в дальнейшем для упрощения мы будем говорить о почтовом сервере под Unix, хотя при обсуждении общей организации системы электронной почты это не имеет принципиального значения).
- Иванов (точнее, пользовательский агент Иванова) может получить доступ к своей почте двумя способами:

- а) Иванов работает на том же компьютере, где находится почтовый сервер. В этом случае MUA Иванова тривиальным образом считывает поступившее сообщение из файла /var/mail/ivanov и сохраняет его, в случае необходимости, куда-то в свой каталог для осуществления своих функций, описанных выше. Этому способу на рисунке соответствует красный пунктир.

- б) Иванов работает на другом компьютере (точнее, Иванов не имеет возможности или желания работать на почтовом сервере). Эта ситуация наиболее типична для пользователей почты в организациях или сообществах. В этом случае для доступа к файлу /var/mail/ivanov через сеть используется протокол POP-3. На почтовом сервере запущена программа POP-сервер, а в MUA Иванова встроен POP-клиент. Этот вариант отражен на рисунке непрерывной красной линией, отходящей от почтовых ящиков.

- Подробно протокол POP-3 рассмотрен в соответствующем пункте ниже. Так как протокол POP-3 работает поверх TCP/IP, нет никаких ограничений на местоположение компьютера Иванова (красная линия на рисунке, исходящая от POP-сервера, в общем случае проходит через Интернет).

- В настоящее время получает распространение протокол IMAP-4, по существу являющийся расширенной версией протокола POP-3. Он, в частности, позволяет пользовательскому агенту каталогизировать и хранить сообщения на почтовом сервере, а не на компьютере пользователя, как это происходит при использовании POP-3. Это удобно в случае, когда Иванов не имеет постоянно закрепленного за собой компьютера - например, Иванов - студент, работающий с почтой из компьютерного класса.

- Теперь рассмотрим исходящее сообщение (сиреневые стрелки) от ivanov@cts.vvsu.ru к bg@aquarium.ru. Сообщение поступает к транспортному агенту двумя способами в зависимости от того, где работает Иванов. Если Иванов работает на почтовом сервере, то его MUA напрямую обращается к транспортному агенту и передает ему сообщение для БГ (сиреневый пунктир). Если же Иванов работает на другом компьютере, то его MUA связывается с транспортным агентом через сеть по протоколу SMTP.

- (Опять, так как протокол SMTP работает поверх TCP/IP, нет никаких ограничений на местоположение компьютера Иванова - сиреневая линия на рисунке, исходящая от комьютера Иванова, в общем случае проходит через Интернет.)

- Получив сообщение, MTA анализирует его заголовок и определяет, что это сообщение направлено в другой почтовый домен и не попадает ни под какие особые случаи (например, не должно быть доставлено через UUCP или отослано по факсу - это все определяется конфигурацией MTA).

- Следовательно, для доставки этого сообщения выбирается агент SMTP, при этом MTA делает запрос в DNS на предмет того, кто является обработчиком почты для домена aquarium.ru. (DNS вернет relay.rinet.ru, IP-адрес=195.54.192.35). Этот адрес вместе с текстом сообщения будет передан агенту доставки, который по протоколу SMTP соединится с указанным адресом и таким образом отправит сообщение транспортному агенту сервера relay.rinet.ru.

- Если во время этой операции произошла нефатальная ошибка (например, удаленный сервер временно выключен), то агент SMTP вернется со статусом "Отложено" и MTA поставит сообщение в очередь для повторной отправки.

- Если запись MX для почтового домена получателя не найдена в DNS, будет сделана попытка найти запись типа A (IP-адрес) для того же доменного имени, т.е. делается предположение, что почтовый домен является именем реального компьютера и на этом компьютере запущен MTA - это справедливо для большинства Unix-систем. Агент доставки попытается доставить сообщение непосредственно на этот адрес.

Почтовые агенты в различных ОС

- В ОС Unix транспортным агентом является программа sendmail, ставшая де-факто стандартом MTA. Кроме того, в программу sendmail входит агент доставки SMTP. Локальный агент доставки - программа mail с ключом "-d". В качестве MUA могут использоваться mail, pine, различные MailTools под X-Windows и другие программы.

- В качестве POP-сервера может быть использована программа qpopper. Все вышеперечисленные программы распространяются свободно, либо являются частью поставки операционной системы.
- MUA со встроенным POP-клиентом (Unix, Windows)- Netscape, Eudora, The Bat и др.

- Под управлением ОС Windows работают такие почтовые серверы как Netscape Messaging Server и Microsoft Exchange. Они администрируются через оконный web-интерфейс, в котором интегрированы все необходимые функции: собственно транспортный агент, POP3-сервер, система администрирования почтовых ящиков пользователей, псевдонимов, групп и списков рассылки. Однако по сравнению с sendmail такие серверы громоздки, не надежны, малопрозрачны и не обладают той степенью гибкости и универсальности, какую имеет sendmail.

Структура email-сообщения

- Базовая структура сообщения электронной почты определена в RFC-822. Сообщение состоит из заголовков и тела сообщения. Заголовки отделяются от тела сообщения пустой строкой.
- Каждый заголовок начинается с новой строки и состоит из ключевого слова, за которым следует двоеточие, и данных:

From: "Sidorov" <sidorov@vvsu.ru>
Если длина данных превышает одну строку,
то последующие строки, относящиеся к этому
же заголовку, начинаются с табуляции:
Received: from u2.farm.idt.net
 (root@u2.farm.idt.net [169.132.8.11]) by
m.vvsu.ru (8.9.1/8.9.1) with ESMTP id MAA00238 for
<sidorov@vvsu.ru>;
Wed, 5 Jan 2000 12:02:28 +1000 (VVO)

- Тело сообщения представляет собой текст в узком смысле (см. выше). Однако, тело сообщения может содержать и символы из расширенного набора (с установленным битом номер 7) - например, кириллицу, - если все агенты, работающие с сообщением, поддерживают восьмибитные символы. В настоящее время большинство используемых агентов такую поддержку обеспечивают.

- Изначально электронная почта предназначалась для пересылки только текстовых сообщений. Для пересылки двоичного содержимого двоичные данные специальным образом кодируются и сообщение снабжается дополнительными заголовками и служебной информацией в соответствии со спецификацией MIME, которая будет рассмотрена ниже в этом пункте.

- При пересылке сообщения по протоколу SMTP говорят о третьей части сообщения - конверте. Конверт - это адреса отправителя и получателя (получателей), передаваемые как аргументы команд "MAIL FROM" (от кого) "RCPT TO" (кому) во время SMTP-сеанса (см. также п. STMP). В простейшем случае адреса на конверте и адреса в заголовках "From:" и "To:" совпадают, но это далеко не всегда так.

- Например, если письмо отправлено нескольким получателям в разные почтовые домены (petrov@a.ru, ivanov@b.ru, sidorov@c.ru, sidorenko@c.ru), то отправляющий MTA размножит это письмо и "разложит" его в 3 конверта, по одному конверту на домен. То есть, в SMTP-сеансе с сервером домена a.ru конверт будет содержать только "RCPT TO: petrov@a.ru", а в сеансе с сервером домена c.ru на конверте будет написано два адресата:

RCPT TO: sidorov@c.ruRCPT TO: sidorenko@c.ru
в то время как в заголовке сообщения могут быть перечислены все адресаты (а могут и не быть - если письмо направлено на список рассылки типа my_friends@vvsu.ru, который состоит из вышеназванных адресов; тогда в заголовке будет адрес списка рассылки.
См. также "Псевдонимы, списки рассылки и форвардинг").

- Вообще, переписывание заголовков и формирование конверта зависит от конфигурации траспортного агента и здесь имеется большой выбор разных вариантов и причин для отличия адресов в конверте от адресов в заголовках. Общее правило: конверт содержит информацию, необходимую для доставки сообщения через сеть; заголовки - информацию для транспортного и пользовательского агентов и самого пользователя.

Заголовки почтового сообщения

Ниже рассмотрены распространенные заголовки, кроме заголовков, добавленных спецификацией MIME.

From: ivanov@a.ru
отправитель; адрес также может иметь форму "Ivan Ivanov" <ivanov@a.ru>.

Reply-To: real_ivanov@a.ru
адрес, на который следует отправлять ответ на письмо. Если этот заголовок отсутствует, ответ отправляется на адрес, указанный в заголовке "From:".

- **To:** sidorov@vvsu.ru, "Petr Petrov"

<petrov@vvsu.ru>

Основной получатель (получатели).

Cc: "Jonh Smith" john@smith.a.comjohn@smith.a.com, sidorenko@c.ru

дополнительный получатель (получатели), если необходимы. При доставке письма для адресов в заголовках "To:" и "Cc:" выполняются одинаковые действия; различия между "To:" и "Cc:" в техническом плане нет.

- **Bcc:** "Fox Mulder" mulder@fbi.gov
- получатель (получатели), невидимый для остальных получателей, если требуется. То есть те, кто перечислен в "To:" и "Cc:", не будут знать, что копия письма отправлена Малдеру.
- **Subject: Happy new Year!**
- тема письма (может отсутствовать); транспортными агентами и агентами доставки не интерпертируется; может интерпертироваться пользовательским агентом в целях фильтрации и сортировки.

- **Date:** Sat, 15 Jan 2000 17:25:32 +1000
- время отправки письма.
- **Message-ID:** 3.0.6.32.20000104175623.007badf0@mail.a.ru
- уникальный идентификатор сообщения, генерируемый MTA-отправителем; для восприятия человеком не предназначен.

- Received: ……
- заголовок "Received:" добавляется каждым транспортным агентом, через которого проходит сообщение, содержит информацию кем, от кого, когда и каким образом получено сообщение.
- В большинстве писем встречаются заголовки, начинающиеся на "X-", это дополнительные заголовки, не определяемые стандартом. Например заголовок "X-Mailer:" содержит информацию о пользовательком агенте, отправившем письмо.

- При пересылке (форвардинге) сообщения другому получателю в заголовки могут быть добавлены поля с префиксом "Resent-" ("Resent-From:", "Resent-To:", "Resent-Date" и т.п.). Эти поля содержат информацию, вставленную тем, кто произвел форвардинг. Например, поле "From:" содержит адрес первоначального отправителя, а "Resent-From:" - адрес того, кто переслал это сообщение. При таком способе форвардинга тело сообщения не изменяется, только добавляются заголовки. Другой метод форвардинга описан в следующем пункте при обсуждении типа данных message/rfc822.

MIME

- MIME (*Multipurpose Internet Mail Extensions - многоцелевые расширения почты Интернет*) - спецификации, определяющие дополнения в формате почтовых сообщений для
- пересылки восьмибитных текстов и полностью двоичных данных;
- поддержки сложных сообщений, состоящих из нескольких разделов (возможно, содержащих данные разных типов).

- Формирование и разбор сообщений в соответствии со спецификациями MIME производится пользовательскими почтовыми агентами. Описание MIME содержится в RFC 2045-2049.

- Для выполнения указанных задач вводятся дополнительные заголовки "***Content-Type:***" и "***Content-Transfer-Encoding:***", которые определяют соответственно тип данных, содержащихся в сообщении (разделе сообщения), и способ представления этих данных, и заголовок "***MIME-Version: 1.0***". Он указывает версию MIME (в настоящий момент 1.0) и используется для обозначения того, что настоящее сообщение является не простым email-сообщением согласно RFC-822, а составлено по спецификации MIME.

- Заголовок "Content-Type:" имеет формат:
- Content-Type: *тип/подтип* [; *параметр=значение* [;...]]
- Параметр (параметры) для некоторых типов данных должны присутствовать, для прочих - необязательны. Неопознанные параметры игнорируются.

Основные типы данных (MIME-types)

- Типы и подтипы, начинающиеся с "x-", не входят в стандарт и считаются определяемыми пользователем для своих нужд, но фактически среди них есть много широко рапространенных наименований, например "audio/x-realaudio". MIME-types используются не только в электронной почте, но и в WWW - для определения типа данных, пересылаемых между сервером и броузером.

- ***text***
- Текстовые данные (в том числе восьмибитные); наиболее распространенные подтипы: plain (обычный текст) и html. Возможный параметр: "charset=*название_кодировки_символов*"; наличие параметра необязательно. (Примеры кодировок символов: us-ascii [текст в узком смысле, только семибитные символы], кириллица: koi8-r, windows-1251, iso8859-5.) Если не указан charset, считается, что это us-ascii.

Пример заголовка:
Content-Type: text/plain; charset=koi8-r
Если заголовок "Content-Type:" отсутствует, то считается, что это
Content-Type: text/plain; charset=us-ascii

- ***Image***
- Неподвижные изображения; примеры подтипов: jpeg, gif. Пример заголовка (параметры необязательны):
- Content-Type: image/jpeg; name="portrait.jpg”
- ***audio***
- Звук; примеры подтипов: mpeg, x-realaudio. Пример заголовка (параметры необязательны):
- Content-Type: audio/x-realaudio; name="song.ra”

- ***Video***
- Видео; примеры подтипов: mpeg, quicktime. Пример заголовка (параметры необязательны):
- Content-Type: video/mpeg; name="movie.mpeg”

- ***application***
- Двоичные данные (поток байт), в общем случае предназначенные для какой-то прикладной программы и не попадающие в вышеперечисленные категории. Подтип позволяет определить, для какой именно программы предназначены данные. Определено большое число различных подтипов (например, postscipt, msword, zip, x-javascript). Если неизвестно, как интерпретировать данные, используется общий подтип octet-stream. Пример заголовка (параметры необязательны):
- Content-Type: application/msword; name="my_file.doc”

- ***multipart***
- Составное сообщение - письмо состоит из нескольких разделов, каждый из которых имеет свои заголовки и тело. Наиболее распространенный подтип - mixed, означающий, что каждый раздел письма может содержать данные любого зарегистрированного типа. Разделы отделяются друг от друга с помощью некоторого уникального набора символов, который указывается в заголовке как значение обязательного параметра "boundary"; выбор этого набора символов с учетом уникальности - задача пользовательского агента, формирующего сообщение.

- Пример заголовка:
- Content-Type: multipart/mixed; boundary="----------CED5632469”
- При разграничении разделов в теле сообщения, значение boundary предваряется двумя минусами (т.е. в данном примере вместо 12 минусов будет 14). Каждый раздел состоит из заголовков и тела (данных); заголовки отделены от данных пустой строкой. Заголовки раздела - "Content-Type:" и "Content-Transfer-Encoding:" - определяют тип данных раздела и их представление

- ***Message***
- Составное сообщение - тело сообщения в свою очередь является email-сообщением, которое может состоять из одного или нескольких частей. Такой тип может применяться при пересылке (форвардинге) сообщений. Основной подтип: rfc822; пример заголовка:
- Content-Type: message/rfc822
- Тип данных message может применяться и в заголовке "Content-Type:" раздела multipart-сообщения; это означает, что тело раздела представляет собой email-сообщение

- Заголовок "Content-Transfer-Encoding:" определяет представление данных в теле сообщения (раздела). Возможные значения:
- ***7bit*** Текст в узком смысле (us-ascii). Данные помещаются в тело собщения (раздела) как они есть.
- ***8bit*** Восьмибитный текст (например, кириллица). Данные помещаются в тело собщения (раздела) как они есть.

- ***binary***
- Двоичные данные (поток байт); помещаются в тело собщения (раздела) как они есть. Это представление обычно не используется, так как хотя современные реализации протокола SMTP и пропускают восьмибитные символы (т.е. фактически байты с любыми значениями кроме нулевого), но все равно требуется чтобы данные состояли из строк не длиннее 1000 символов - т.е. чтобы комбинация CRLF встречалась не реже чем каждые 998 символов. (Это как раз и есть представление "8bit".)

- ***quoted-printable***
- Восьмибитный текст в закодированном виде. Кодировка выполняется посимвольно. Одни символ кодируется следующим образом (алгоритм неполный, см. RFC2045 для деталей): символ с ASCII-кодом от 32 до 127 включительно передается как он есть; символ с другим кодом передается в виде тройки символов, состоящей из символа "=", за которым следует шестнадцатеричное значение кода символа. Символ "=", являющийся частью текста, кодируется как "=3D". Переводы строки в конце каждой строки текста передаются как они есть

- Предполагается что это кодировка используется для текстов, которые состоят в основном из символов латиницы, цифр и знаков препинания, с незначительными вкраплениями других символов.

- ***Base64***
- Произвольные двоичные данные, закодированные по алгоритму base64. Сущность алгоритма в следующем (описание и таблицу символов см. в RFC2045): из входного потока берется 24 бита (3 октета), которые разбиваются на 4 группы по 6 бит в каждой. Каждое возможное значение 4-битной группы (0-63) соответствует одному символу из таблицы 7-битных текстовых символов (0='A', 1='B', ..., 26='a', 27='b', ..., 52='0', ..., 62='+', 63='/').

- Таким образом 3 входных октета с произвольными значениями преобразуются на выходе в четыре семибитных символа, которые без проблем обрабатываются любыми почтовыми агентами. Получившийся выходной поток символов разбивается на строки длиной не более 76 символов.
- Можно разрабатывать и использовать также нестанданртные представления; их наименования обязаны начинаться с символов "x-"
-

- **Примеры почтовых сообщений с заголовками**
- 1. Сообщение, состоящее из двух частей. Первая часть - текст на русском языке в кодировке КОИ-8; при создании письма этот текст был введен как собственно текст письма. Вторая часть создана почтовой программой как attachment, содержащий файл test.jpg.

- Файл test.jpg, названный так умышленно, состоит из шести символов "abcdef"; тем не менее почтовая программа, основываясь на расширении, интерпретировала его как JPEG-файл, выставила соответствующий Content-Type и провела кодировку содержимого как двоичных данных по base64 (получилось "YWJjZGVm").

- Received: from ada.vvsu.ru (ada.vvsu.ru [212.16.195.70])
  by maria.vvsu.ru

(8.8.3/8.8.3) with SMTP id RAA04870 for
  fire@maria.vvsu.ru;
  Tue, 18 Jan 2000 17:15:26 +1000 (VVO)

Message-ID: <388412B0.3522@vvsu.ru>

Date: Tue, 18 Jan 2000 17:13:53 +1000

From: Maxim Mamayev m2@vvsu.ru

X-Mailer: Mozilla 3.0 (Win95; I)

MIME-Version: 1.0

To: fire@maria.vvsu.ru

Subject: test of mixed message

Content-Type: multipart/mixed; boundary="simple
  boundary”

This is a multi-part message in MIME format.

- Эта часть (преамбула) игнорируется в MIME-сообщениях
- Как правило пользовательский агент помещает сюда объявление о том, что это MIME-сообщение на случай, если агент получателя
- не поддерживает MIME.
- --simple boundary
- Content-Type: text/plain; charset=koi8-r
- Content-Transfer-Encoding: 8bit

- Основной текст сообщения на русском языке в КОИ-8
- --simple boundary
- Content-Type: image/jpeg; name="test.jpg“
- Content-Transfer-Encoding: base64
- Content-Disposition: inline; filename="test.jpg“

YWJjZGVm

--simple boundary

Это эпилог. Он игнорируется как и преамбула.

- 2. Сообщение, являющееся результатом пересылки (форвардинга) некоторого исходного сообщения. Исходное сообщение (от xhawk@chat.ru для m2@vvsu.ru) содержится внутри тела нового сообщения
- Received: from ada.vvsu.ru (ada.vvsu.ru [212.16.195.70]) by maria.vvsu.ru
- 8.8.3/8.8.3) with SMTP id UAA04969 for <fire@maria.vvsu.ru>;
- Tue, 18 Jan 2000 20:54:30 +1000 (VVO)
- Message-ID: 3884460A.5AD0@vvsu.ru
- Date: Tue, 18 Jan 2000 20:52:58 +1000
- From: Maxim Mamayev <m2@vvsu.ru>

- X-Mailer: Mozilla 3.0 (Win95; I)
- MIME-Version: 1.0
- To: fire@maria.vvsu.ru
- Subject: [Fwd: forward it]
- Content-Transfer-Encoding: 8bit
- Content-Disposition: inline
- Content-Type: message/rfc822

- Received: from chat.ru (surf-6.vvsu.ru [212.16.195.86]) by wildcat.vvsu.ru
- (Netscape Messaging Server 3.62) with ESMTP id 372
- for <m2@vvsu.ru>; Mon, 27 Dec 1999 17:43:13 +1000
- Message-ID: 386718EE.90BD95DF@chat.ru
- Date: Mon, 27 Dec 1999 17:44:46 +1000

- From: Yankee xhawk@chat.ru
- X-Mailer: Mozilla 4.51 [en] (Win95; I)
- MIME-Version: 1.0
- To: m2@vvsu.ru
- Subject: forward it
- Content-Type: text/plain; charset=koi8-r
- Content-Transfer-Encoding: 8bit
- Текст исходного сообщения

- 3. Сообщение, являющееся результатом форвардинга сообщения из примера номер 1 (которое в свою очередь является сложным сообщением, состоящим из двух частей). Более того, при форвардинге в "объемлющее" новое сообщение был добавлен некоторый текст.

Received: from ada.vvsu.ru (ada.vvsu.ru [212.16.195.70]) by maria.vvsu.ru
(8.8.3/8.8.3) with SMTP id SAA04920 for <fire@maria.vvsu.ru>;
Tue, 18 Jan 2000 18:58:00 +1000 (VVO)Message-ID:
38842ABB.39CC@vvsu.ru
Date: Tue, 18 Jan 2000 18:56:27 +1000
From: Maxim Mamayev <m2@vvsu.ru>X-Mailer: Mozilla 3.0 (Win95; I)
MIME-Version: 1.0To: fire@maria.vvsu.ruSubject: [Fwd: test mixed]
Content-Type: multipart/mixed;
boundary="----------20EADB04695"

This is a multi-part message in MIME format.
----------20EADB04695
Content-Type: text/plain; charset=koi8-r
Content-Transfer-Encoding: 8bit
Текст, добавленный при форвардинге
----------20EADB04695
Content-Type: message/rfc822Content-Transfer-Encoding: 8bit
Content-Disposition: inline
Message-ID: 388412B0.3522@vvsu.ru
Date: Tue, 18 Jan 2000 17:13:53 +1000
From: Maxim Mamayev m2@vvsu.ru
X-Mailer: Mozilla 3.0 (Win95; I)
MIME-Version: 1.0
To: fire@maria.vvsu.ru
Subject: test mixed
Content-Type: multipart/mixed; boundary="simple boundary"

- This is a multi-part message in MIME format.
- --simple boundary
- Content-Type: text/plain; charset=koi8-r
- Content-Transfer-Encoding: 8bit
- Текст исходного сообщения
- --simple boundary
- Content-Type: image/jpeg; name="test.jpg“
- Content-Transfer-Encoding: base64
- Content-Disposition: inline; filename="test.jpg“
- YWJjZGVm
- --simple boundary
- ----------20EADB04695--

Основные команды протокола SMTP

- Для пересылки почтовых сообщений через Интернет между транспортными агентами и от MUA к MTA (см. рис. 4.1) используется протокол SMTP (Simple Mail Transfer Protocol).
- Номер порта сервера SMTP - TCP/25. После установления соединения с клиентом сервер ожидает ввода команд и данных в текстовом виде. Строчные и прописные буквы в командах не различаются. Реакция сервера заключается в трехзначном числовом коде, снабженном текстовым комментарием.

- Числовой код предназначен для автоматической обработки ответов сервера программой-клиентом. Код, начинающийся на 2, является положительным ответом, на 3 - промежуточным положительным откликом (т.е. после ввода команды ожидаются дополнительные данные), коды вида 5xx сигнализируют об ошибке.
- Основные команды SMTP:

- Вкратце необходимо отметить следующее. Выдача блоков адресов потребителям производится локальными Интернет регистратурами (LIR), которые, в свою очередь, получают их от региональных регистратур. В Европе это – бесприбыльная организация RIPE (http://www.ripe.net/), финансируемая провайдерами. Основные функции региональных регистратур – координация использования IP адресов и, соответственно, маршрутизации в регионе. Для получения своего блока публичных адресов необходимо заполнить соответствующие формы RIPE и направить их вашему LIR.

- Основные команды SMTP:
- HELO *hostname*
- - первая команда сеанса, *hostname* - доменное имя вызывающего хоста (клиента).
- MAIL FROM: *email_адрес_от_кого*
- - обратный адрес.
- RCPT TO: *email_адрес_кому*
- - адрес получателя (в случае нескольких адресатов команда повторяется для каждого адресата).

- DATA
- - начало ввода текста сообщения; сервер посылает промежуточный положительный отклик 354 и рассматривает все последующие строки в качестве текста (тела) сообщения; конец ввода - строка состоящая из одной точки. Перед текстом сообщения вводятся поля заголовка (см. выше п. Структура email-сообщения). Каждое поле заголовка должно начинаться с новой строки. Между заголовком и текстом сообщения должна быть ***одна пустая строка***.

- VRFY *email_адрес*

- - выдается положительный отклик (250,251 или 252), если сервер может ***попытаться*** доставить сообщение по указанному адресу; иначе выдается отрицательный отклик 550. Если *email_адрес* - не локальный адрес на сервере, то положительный отклик не обязательно означает, что этот адрес существует. Для локальных адресов производится подстановка из файла /etc/aliases (если адрес там указан) и в поле текстового комментария выводится результат, часто к нему добавляется настоящее имя пользователя.

- EXPN *email_addr*
- - если *email_addr* - локальный адрес списка рассылки, то вывести адреса в этом списке; иначе поведение команды не определено. Как правило, если *email_addr* - локальный адрес, то выполняется подстановка из файла /etc/aliases (если там этот адрес указан); иначе просто выдается положительный отклик 250.

- RSET
- - сброс сеанса к начальному состоянию (как после ввода HELO).
- QUIT
- - конец связи.
- Команды EXPN и VRFY не обязательно выполняются сервером; часто их отключают из соображений секретности. Для передачи почты эти команды не нужны; фактически, они предназначены для человека. Действия этих команд в стандарте четко не определены и их реализация не является обязательной.

- Сущетсвуют также дополнительные команды - так называемый Расширенный SMTP (ESMTP). Не все серверы поддерживают команды ESMTP и каждый сервер может поддерживать какое-то свое подмножетсво ESMTP-команд, в том числе нестандартных. Для того, чтобы выяснить, какие дополнительный команды поддерживаются, следует вместо HELO подать команду EHLO.

- **Основные команды протокола POP-3**
- Номер порта сервера POP - TCP/110. После установления соединения с клиентом сервер ожидает ввода команд и данных в текстовом виде. Строчные и прописные буквы в командах не различаются. Реакция сервера - строка начинающаяся с метки "+OK" или "-ERR", за которой следует текстовый комментарий и, если команда это подразумевает, с новой строки выводятся данные (текст сообщения или листинг сообщений). Вывод данных заканчивается строкой, содержащей только символ "." ("точка"). Если среди данных есть такая строка, то точка в этой строке удваивается.

- Команды POP-3:
- USER *имя_пользователя*
- - первая команда сеанса, вводится имя пользователя (идентификатор почтового ящика).
- PASS *пароль*
- - вторая команда сеанса, вводится пароль.
- STAT
- - после метки "+OK" выводит два числа: число сообщений и их общий объем в байтах.
- LIST *n*
- - если *n* указано, то после метки "+OK" выводит размер сообщения номер *n* в байтах. Иначе выводит список из двух колонок: номер сообщения, пробел, размер сообщения в байтах; вывод списка заканчивается строкой, содержащей только символ "." ("точка").

- RETR *n*
- - выводит сообщение номер *n*. Вывод заканчивается строкой, содержащей только символ "." ("точка").
- DELE *n*
- - удаляет сообщение номер номер *n* с сервера; при этом нумерация сообщений не изменяется, а все удаленные *в данном сеансе* сообщения могут быть восстановлены командой REST.
- LAST
- - после метки "+OK" выводит номер последнего по времени сообщения, для которого была выполнена команда RETR; эта информация сохраняется между сеансами, что позволяет не запрашивать дважды уже полученные пользователем, но не удаленные с сервера сообщения.

- TOP *n m*
- - выводит заголовок и *m* первых строк сообщения номер *n*. Вывод заканчивается строкой, содержащей только символ "." ("точка").
- RSET
- - отменяет удаление всех сообщений, удаленных в данном сеансе.
- NOOP
- - нет операции.
- QUIT
- - конец связи.

Литература

- В. Амато. Основы организации сетей Cisco.
- В.Г. Олифер, Н.А. Олифер. Основы сетей передачи данных.
- Ю.В. Новиков, С.В. Кондратенко. Основы локальных сетей.