Билет № 1.
Предпосылки появления сетей ЭВМ и развития информационных технологий.

Две главные предпосылки появления сетей ЭВМ и развития информационных технологий: *создание технически сложных систем* и *необходимость быстрого получения, накопления и обработки информации*.
Создание технически сложных систем потребовало принципиально новых подходов к их проектированию. Технически сложную систему не всегда можно представить в натуральном виде. Требуются моделирование, сложные расчеты, вычислительные схемы, специальные методы борьбы со сложностью и создаваемого изделия, и его модели.
Как убедиться, что создаваемая система при любых обстоятельствах будет функционировать корректно? Новые требования к скоростям расчетов, алгоритмам, численным методам, средствам моделирования. Уже в 1940-х гг. - распараллеливание. При этом без моделирования и предварительного численного анализа создавать сложные технические системы было уже невозможно.
Э.Ферми и проект ≪Манхеттен≫
Просто создать объект уже мало: его надо уметь поддерживать в работоспособном состоянии, выявлять неисправности, ремонтировать при выходе из строя, модифицировать и развивать в ходе эксплуатации, т.е. эксплуатация технически сложных систем потребовала специальной организации работ, где без информационных технологий обойтись нельзя.

Помимо технических предпосылок определяющую роль в развитии информационных технологий сыграли предпосылки социальные. Прежде всего, надо помнить, что информации всегда не хватало, и стоила она дорого.
В обществе XX в. налицо были следующие тенденции: демографический рост; концентрация капитала, индустриализация развитых стран и
децентрализация управления; рост числа людей, вовлекаемых в процессы принятия решений. При развитии данных тенденций отсутствие эффективных методов коммуникации, распределённого доступа к информации, автоматизации ее сбора, обработки и хранения тормозили развитие индустриализации как на внутригосударственном, так и на межгосударственном уровнях.
Скорость принятия решения можно представить в виде числового значения и измерить в стоимостном выражении. Необходимо повышать эту скорость. Необходимыми стали новые быстрые средства коммуникации. Появились телеграф, телефон, радио, телевидение. Постепенно возникла необходимость интеграции этих видов связи. Важной тенденцией наших дней является то, что телефонные, телевизионные, радио- и спутниковые сети связи, используемые в технологиях сбора, передачи, обработки и хранения информации, начинают сливаться и интегрироваться в единую сеть.

Интеграция компьютеров со средствами передачи данных коренным образом изменила представление об организации вычислительных систем. Появились сети ЭВМ.
Появление и развитие сетей ЭВМ и средств коммуникации были востребованы военными и промышленностью. Военным требовались падежные живучие системы коммуникации, которые невозможно уничтожить однократным ударом, а также надежные высокоскоростные системы сбора и обработки информации. Промышленности требовались средства, обеспечивающие эффективное управление производством, развитие и расширение рынка.
Сети ЭВМ стали основой информационных технологий. Сегодня сети ЭВМ во многом определяют скорость адаптации компании к изменяющимся внешним условиям и ее способность быстро и экономически эффективно управлять своей деятельностью.
Не меньшее значение информационные технологии имеют и для отдельной личности. Ярким примером является их применение для развития различных форм и методов обучения людей. Следует отметить, что в современных условиях изменилась роль и значение образования в деятельности любого государства. Рост интеллектуализации рабочих мест. Промышленность не может работать без притока образованных специалистов. Если раньше основной функцией образования было развитие личности, то теперь — это обеспечение промышленности профессиональными кадрами. Унификация образовательных стандартов в разных странах.
Таким образом, конкурентоспособность как отдельного предприятия, так и государства в целом напрямую теперь зависит от развития информационных технологий.

Билет № 2.

Основные движущие силы развития информационных технологий (закон Мура и закон Гилдера).

Состояние и направление развития информационных технологий определяют три основные составляющие: *микроэлектроника*, *телекоммуникации* и *инженерия программного обеспечения*.

1946 – год создания первого компьютера ENIAC. Д. Моучли, Д. Эккерт.
Вес - 27 т, 18 000 электронных ламп, 1 500 реле, потребляла около 150 кВт энергии.
1947 – точечный транзистор. В.Шокли, Д. Барден и У. Бретейн. 1956 г. - Нобелевская премия по физике.

1971 - первый микропроцессор Intel 4004, частота 108 кГц и содержал 2 300 транзисторов.
1978 - микропроцессор Intel 8086 - 29 000 транзисторов, частота 5 МГц
2001 - микропроцессор Pentium 4 - частота 1,7 ГГц, число транзисторов 42 млн.
С 1971 г. тактовая частота процессоров Intel возросла в 28 тыс. раз, т.е. со 108 кГц до 3 ГГц, а среднее число транзисторов в одном процессоре выросло в 350 тыс. раз.
Тенденция увеличения числа транзисторов на кристалле была сформулирована в 1965 г. Г. Муром (одним из основателей Intel) в виде закона, который носит его имя:
количество транзисторов в интегральной схеме с минимальной ценой удваивается каждые 18 месяцев.

И сам компьютер, и его интерфейс с человеком подверглись существенным изменениям. Персональный компьютер впервые появился в 1981, и за следующие 20 лет он совершил революцию: до этого он был доступен только специалистам, а теперь он «вошел» почти в каждый дом, с ним работают школьники, он стал мобильным. В настоящее время в мире функционирует более 1,5 млрд ПК. Отдельный класс устройств на основе микропроцессоров составляют встроенные системы. Это весьма широкий класс устройств: от бытовых приборов до сложных технических систем. Уже в 2002 г. устройства, не являющиеся персональными компьютерами, составляли около 50 % от всех устройств доступа в Интернет, а к 2005 г. число таких устройств превысило число ПК. К 2005 г. На одного жителя Европы и Северной Америки в среднем приходилось около 20 микропроцессов, размещенных в бытовых приборах. Сегодня микропроцессор можно встретить везде: в кроссовках, кастрюлях, автомобилях, самолетах, на кораблях.

В настоящее время, с одной стороны, компьютеры объединяют в сети, оборудуя их необходимыми средствами телекоммуникации, а с другой стороны, традиционные средства коммуникации, такие как телефон, превращают из простого средства передачи голоса в изощренные средства интерактивного беспроводного взаимодействия.
В области телекоммуникаций действует закон Гилдера, выявивший на основании статистических данных следующую тенденцию:
пропускная способность телекоммуникационных каналов удваивается каждые шесть месяцев.
Современная система передачи данных, или просто кабель, за секунду способна пропустить столько данных, сколько в 1997 г. пропускал весь Интернет. Так, например, в 1997 г. Ethernet обладал пропускной способностью в 100 Мбит, а сегодня это 100 Гбит.

Билет № 3.
Кто, как и для чего использует Сеть: интранет.

В сфере бизнеса можно выделить области применения сетей в следующих направлениях:
• интранет — использование сети для управления и производственных нужд внутри предприятия;
• B2B (Business To Business) — взаимодействие с другими предприятиями, которые, в свою очередь, можно подразделить на взаимодействие с предприятиями-заказчиками, взаимодействие с предприятиями-поставщиками и создание виртуальных предприятий;
• B2C (Business To Customer) — взаимодействие предприятия с конечными пользователями их продукции;
• электронное правительство, которое, в свою очередь, разделяется на B2G (Business To Goverment) — взаимодействие предприятия с государством и G2C (Government To Citizen) — взаимодействие государства с гражданами;
• C2C (client-To-client) — набор услуг (технологий), которые позволяют клиентам – физическим лицам обмениваться товарами или оказывать взаимные услуги.

Интранет — это средства, обеспечивающие и регулирующие доступ сотрудников компании к внутрикорпоративным средствам коммуникации, информационным и вычислительным сервисам. Упрощенно интранет — это внутренняя корпоративная сеть, построенная на интернет-технологиях. Не все пользователи находятся в зоне сети интранет. Некоторые из них подключаются к корпоративным информационным ресурсам извне.
Интранет обеспечивает для всех сотрудников единый способ доступа к информации, единую унифицированную среду работы, единый формат документов. Такой подход позволяет сотрудникам наиболее эффективно использовать накопленные корпоративные знания, оперативно реагировать на происходящие события, где бы они ни находились, а предприятию в целом предоставляет новые возможности организации своего бизнеса.
Интерфейс доступа в интранет внешне имеет вид web-сайта (а точнее портала), который кроме обычных функций может идентифицировать сотрудника компании, отличить его от простого пользователя сетью и открыть доступ к тем информационным и вычислительным внутрикорпоративным сервисам, с которыми ему разрешено работать. К типовым сервисам относятся: средства планирования и управления временем сотрудника (календарь), адресная книга, средства передачи сообщений и другие средства, с помощью которых можно связаться с другими сотрудниками. Важной информационной услугой, осуществляемой через интранет, является доступ сотрудника к системе управления знаниями предприятия. Эта система призвана аккумулировать и хранить информацию и знания, возникающие в голове каждого сотрудника. Система управления знаниями предприятия позволяет так организовать работу сотрудников, чтобы они сами систематически пополняли ее своими знаниями, таким образом знания остаются на предприятии и после ухода сотрудника.
Важными характеристиками интранета являются открытость и масштабируемость. Интранет должен позволять наращивать функциональность и интегрировать информационные прикладные системы организации. Это свойство позволяет предприятию развивать информационные системы эволюционным путем по мере возникновения необходимости.

Билет № 4.
Кто, как и для чего использует Сеть: B2B.

В сфере бизнеса можно выделить области применения сетей в следующих направлениях:
• интранет — использование сети для управления и производственных нужд внутри предприятия;
• B2B (Business To Business) — взаимодействие с другими предприятиями, которые, в свою очередь, можно подразделить на взаимодействие с предприятиями-заказчиками, взаимодействие с предприятиями-поставщиками и создание виртуальных предприятий;
• B2C (Business To Customer) — взаимодействие предприятия с конечными пользователями их продукции;
• электронное правительство, которое, в свою очередь, разделяется на B2G (Business To Goverment) — взаимодействие предприятия с государством и G2C (Government To Citizen) — взаимодействие государства с гражданами;
• C2C (client-To-client) — набор услуг (технологий), которые позволяют клиентам – физическим лицам обмениваться товарами или оказывать взаимные услуги.

B2B — условное обозначение набора услуг, которые одна фирма может оказать другой, используя сеть Интернет. На сегодня сформировались два основных направления в этой сфере: интернет-биржи и интернет-консалтинг. Интернет-биржи являются несколько расширенным представлением обычных бирж с тем только отличием, что за счет больших возможностей по отображению визуальной информации с их помощью стала реальной торговля не только классическими
биржевыми товарами, такими как нефть, зерно, металлы, но и некоторыми стандартными видами товарной продукции (оборудованием, комплектующими, компьютерной техникой и т.д.)

Интернет-консалтинг в Сети обеспечивает, в первую очередь, поиск и обработку информации из самой Сети (поиск поставщиков и потребителей, мониторинг, анализ рынков и т.д.).
Многие электронные биржи позволяют не только заключать сделки, но и планировать, а также управлять поставками. Управление поставками очень важно, так как позволяет регулировать стоимость конечного продукта. Основной формой взаимодействия в сфере B2B является сотрудничество.

Билет № 5.
Кто, как и для чего использует Сеть: B2C и электронное правительство.

В сфере бизнеса можно выделить области применения сетей в следующих направлениях:
• интранет — использование сети для управления и производственных нужд внутри предприятия;
• B2B (Business To Business) — взаимодействие с другими предприятиями, которые, в свою очередь, можно подразделить на взаимодействие с предприятиями-заказчиками, взаимодействие с предприятиями-поставщиками и создание виртуальных предприятий;
• B2C (Business To Customer) — взаимодействие предприятия с конечными пользователями их продукции;
• электронное правительство, которое, в свою очередь, разделяется на B2G (Business To Goverment) — взаимодействие предприятия с государством и G2C (Government To Citizen) — взаимодействие государства с гражданами;
• C2C (client-To-client) — набор услуг (технологий), которые позволяют клиентам – физическим лицам обмениваться товарами или оказывать взаимные услуги.

B2C — условное обозначение набора услуг, которые фирма может оказать клиенту (конечному потребителю), используя возможности, предоставляемые Сетью. Типичными представителями данного направления применения Сети являются интернет-магазины. Безусловно, к этому классу приложений относятся банковские услуги, разнообразные информационные и справочные услуги, ориентированные на конечного потребителя, туристические услуги, сетевые библиотеки игр, книг, фильмов, услуги по дистанционному обучению.
Клиент может оформить заказ и отследить его выполнение, получая необходимые уведомления о прохождении определенного этапа изготовления, вплоть до даты доставки товара. Развитие данного вида предприятий предопределено резким снижением затрат. Остаются только затраты на склад и службу доставки

Концепция ***электронного правительства*** (Electronic Government) была провозглашена в 1997 г. в США на самом высоком правительственном уровне. Цель программы была заявлена как снижение издержек при финансировании деятельности госаппарата и повышение открытости и прозрачности органов управления на основе внедрения технологии электронной коммерции. Новые тенденции в электронной коммерции ознаменовались появлением аббревиатур B2G, G2C, G2G, обозначающих новые сферы бизнеса, в которые так или иначе вовлечено государство (Government) — Business To Government, Government To Citizens, Government To Government. Государство также включилось в процесс электронизации. Роль электронной коммерции в организации работы государственных органов двояка. С одной стороны, это снижение издержек и экономия средств налогоплательщиков на содержание и финансирование деятельности госаппарата (B2G). С другой стороны, это повышение открытости и прозрачности органов управления, обеспечение свободного доступа граждан ко всей необходимой государственной информации (G2C). Разделение новой индустрии на секторы B2G и G2C чисто функциональное. Однако в обоих секторах необходимую инфраструктуру обеспечивает Сеть. В первую очередь электронными могут стать самые очевидные функции государства — сбор налогов, регистрация транспортных средств, регистрация патентов, выдача лицензий, получение необходимой информации, заключение договоров и оформление поставок необходимых государственному аппарату материалов, оборудования. В результате сокращается объем бумажной работы, и проведение необходимых процедур значительно ускоряется. То, для чего раньше требовалось долгое стояние граждан в очередях, общение с правительственными чиновниками, а также производство и перемещение большою количества бумажных документов будет происходить теперь за несколько минут. B2G может стать эффективным оружием в борьбе с коррупцией.

Билет № 6.
Кто, как и для чего использует Сеть: C2C.

В сфере бизнеса можно выделить области применения сетей в следующих направлениях:
• интранет — использование сети для управления и производственных нужд внутри предприятия;
• B2B (Business To Business) — взаимодействие с другими предприятиями, которые, в свою очередь, можно подразделить на взаимодействие с предприятиями-заказчиками, взаимодействие с предприятиями-поставщиками и создание виртуальных предприятий;
• B2C (Business To Customer) — взаимодействие предприятия с конечными пользователями их продукции;
• электронное правительство, которое, в свою очередь, разделяется на B2G (Business To Goverment) — взаимодействие предприятия с государством и G2C (Government To Citizen) — взаимодействие государства с гражданами;
• C2C (client-To-client) — набор услуг (технологий), которые позволяют клиентам – физическим лицам обмениваться товарами или оказывать взаимные услуги.

C2C — условное обозначение процесса (технологии) client-to-client (потребитель — потребителю), т.е. набор услуг (технологий), которые позволяют физическим лицам обмениваться товарами или оказывать взаимные услуги, не прибегая к помощи посредников за счет использования возможностей Сети.
Основные возможности: обучение, игры, покупки, информационный поиск, получение справочной информации, общение.
Интернет-аукционы - берут на себя функцию технического обеспечения сделки (дать объявление о продаже или запрос на покупку, продемонстрировать товар, оговорить предварительные условия и т.д.) и позволяют клиентам самостоятельно покупать и продавать предметы личной собственности.
Список основных средств общения в Интернете: веб-форумы; блоги; вики-проекты; электронная почта; группы новостей; интернет-радио; интернет-телевидение; Skype; IP-телефония; мессенджеры; FTP-серверы; ICQ; поисковые системы; социальные сети.

Однако Сеть таит и много опасностей.
1. Психологическая зависимость.
2. Всевозможные формы социального хакинга (несанкционированного доступа, атаки на страницы социальных сайтов в целях получения доступа к конфиденциальной информации: паролей к е-почте, электронным денежным ресурсам и пр.). Социальный хакинг в последнее время становится все более распространенным видом разбоя. Формы жульничества в Сети весьма и весьма разнообразны. Серьезную проблему представляет использование Интернета в экстремистских целях.

Билет № 7.

Основные движущие силы развития информационных технологий (инженерия программного обеспечения).

В первые десятилетия компьютерной истории профессионального программирования как такового не было. Программирование рассматривалось как кодирование. Если в качестве примера посмотреть на первые учебники по программированию, то они представляли собой сборники профессиональных рецептов, как ввести число, строку символов, и т.д. Программирование было ремеслом, а не видом индустриальной деятельности. Программы в массе своей не были продуктами.

Только к концу 1960-х гг. стало складываться представление о новой специальности. При этом программирование из кустарного производства стало трансформироваться, с одной стороны, в науку, а с другой — в отрасль промышленности. Пришло осознание того, что программа — это самостоятельный продукт, не зависящий от изготовителя железа, а программирование — это область инженерии.

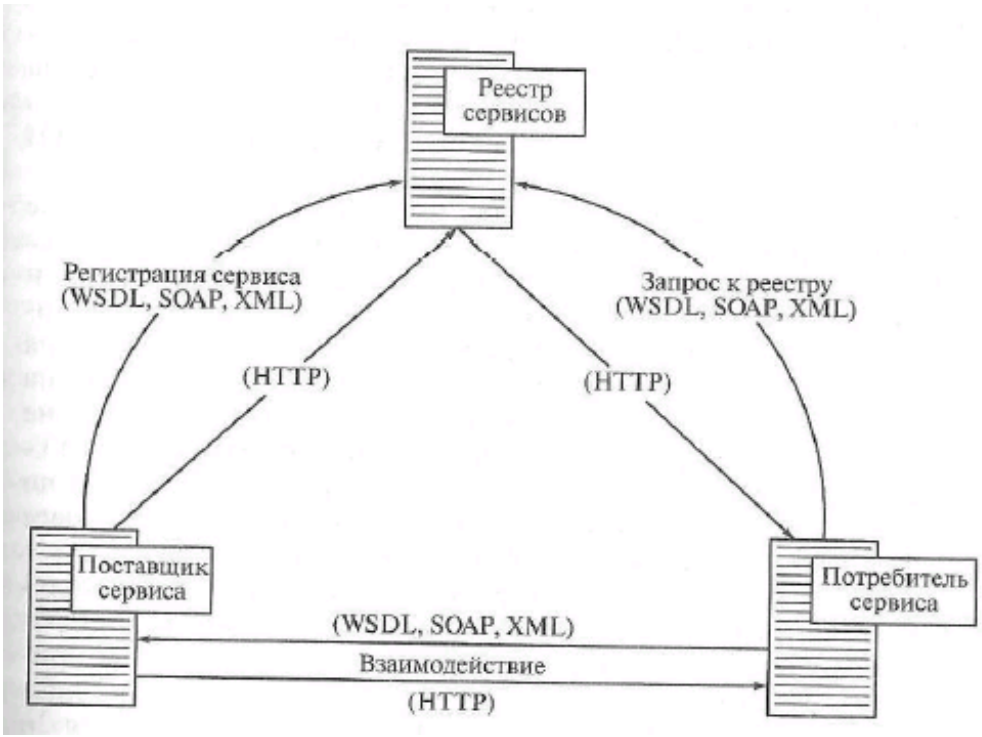
Отправной точкой в истории инженерии программного обеспечения считается конференция, состоявшаяся в 1968 г. в Западной Германии в г. Партенкирхен по инициативе Комитета по науке НАТО. Поводом для нее стало осознание кризисной ситуации, сложившейся к этому времени в области разработки программного обеспечения, суть которой заключалась в том, что сложность программ и мощность компьютеров вступили в противоречие с технологическими возможностями программирования. Закона, постулирующего количественное развитие программного обеспечения и подобного закону Мура, не существует. Однако рост производительности труда программистов отмечают многие. Программирование в 1970-х гг. существенно отличается от программирования в 1990-х или в 2000-х гг. Можно сделать вывод, с одной стороны, о рекордных, невиданно высоких темпах роста производительности труда программистов, а с другой стороны, в сравнении с темпами роста, определяемыми законами Мура и Гилдера, приведенные показатели непропорционально низкие. С уверенностью можно сказать одно, что рост производительности труда программистов представляет собой плохо управляемый процесс: программирование губит иллюзия простоты и вседозволенности.

Уже в 1970-е гг. пришло понимание того, что программы как продукт инженерной деятельности имеют беспрецедентную в истории человеческой цивилизации сложность, а следовательно, нельзя каждый раз разработку программной системы начинать с нуля. Поэтому активно стали развиваться методы программировании, сновная идея которых состояла в использовании при создании программных систем ранее написанных компонентов — кирпичиков, в которых аккумулировался бы опыт предшествующих разработок и корректность работы которых не надо было бы каждый раз обосновывать.

Однако реализация этой идеи потребовала решения череды очень непростых задач, осознание которых во многих случаях приходило не сразу. Так, например, эти компоненты должны были работать в разных операционных средах, быть многократно используемыми в разных программных контекстах, и т.д.

Билет № 8.
Сервис ориентированные архитектуры.

Сервис-ориентированная архитектура – SOA - подход к разработке программного обеспечения, основанный на использовании сетевых сервисов (служб) со стандартизированными интерфейсами.
В самом общем виде SOA предполагает наличие трех основных участников: поставщика сервиса, потребителя сервиса и реестра сервисов, где указаны стандарты языков и протоколов, используемых при реализации SOA. Взаимодействие участников SOA организовано достаточно просто: поставщик сервиса регистрирует свои сервисы в реестре, а потребитель обращается к реестру с запросом. Для использования сервиса необходимо следовать соглашению об интерфейсе обращения к сервису: интерфейс должен не зависеть от среды разработки сервиса. SOA предполагает возможность добавления сервисов, а также их модернизацию. Интерфейс компонентов SOA-программы инкапсулирует, т.е. скрывает детали реализации конкретного компонента от остальных компонентов. Таким образом, SOA предоставляет гибкий и элегантный способ комбинирования и многократного использования компонентов программы для построения сложных распределенных программных комплексов, т.е. компоненты программы могут быть распределены по разным узлам сети и предлагаться как независимые слабосвязанные заменяемые сервисы-приложения.



Программные комплексы, разработанные в соответствии с SOA, часто реализуются как набор веб-сервисов, интегрированных с помощью известных стандартных протоколов. Веб-сервис — это набор логически связанных функций, которые могут быть вызваны удаленно через Интернет. Информация о том, какие функции предоставляет данный веб-сервис, содержится в документе на языке WSDL (Web Services Description Language), а для поиска существующих веб-сервисов предполагается использовать специальные реестры, совместимые со спецификацией UDDI (Universal Description Discovery and Integration).
SOA обеспечивает предприятию высокую скорость адаптации к динамично изменяющимся условиям современного рынка.

Билет № 9.
Модели сетевого взаимодействия OSI ISO и TCP/IP.

Модель OSI имеет уровневую организацию. Она включает в себя семь уровней: физический, канальный, сетевой, транспортный, сессии, представления и прикладной.
Ключевыми в этой модели являются понятия сервиса, интерфейса и протокола. Под сервисом понимают услуги, которые нижерасположенный уровень оказывает по запросам вышерасположенного. В этой модели нижерасположенный уровень свои услуги может предоставлять только вышерасположенному уровню. Интерфейс определяет формирование и передачу запроса на услугу. Активные элементы уровня, т.е. элементы, которые могут сами совершать действия, в отличие от элементов, над которыми совершают действия, называются *активностями.* Активности могут быть программными и аппаратными. Активности одного и того же уровня на разных машинах называются *равнозначными,* или *одноименными.* Активности уровня *n* + 1 являются *пользователями сервиса,* создаваемого активностями уровня n, которые, в свою очередь, называются *поставщиками сервиса.* Сервис может быть разного качества. Правила и соглашения по установлению соединения, его поддержанию и обмену данными по нему между активностями, расположенными на одинаковом уровне на разных машинах, называется *протоколом.*
Доступ к сервису в модели OSI осуществляется через так называемые *точки доступа к сервису* — SAP (Service Access Points), каждая из которых имеет уникальный адрес. Взаимодействие между двумя соседними уровнями в этой модели можно описать следующим образом: активность на уровне *n* + 1 передает интерфейсную единицу данных — IDU (Interface Data Unit) активности на уровне *n* через SAP (рис. 1.3). IDU состоит из сервисной единицы данных — SDU (Service Data Unit) — и управляющей информации. SDU передается далее по сети равнозначной сущности, а затем - - на уровень *n* + 1. Управляющая информация необходима нижерасположенному уровню, чтобы правильно передать SDU, но она не является частью передаваемых данных. Чтобы передать SDU по сети нижерасположенному уровню, может потребоваться разбить ее на части. При этом каждая часть снабжается *заголовком* и *концевиком* и передается как самостоятельная единица данных протокола PDU (Protocol Data Unit). Заголовок в PDU используется протоколом при передаче. В нем указывается, какой PDU содержит управляющую информацию, а какой — данные, порядковый номер PDU и т.д. Формально сервис можно описать в терминах примитивных операций, или *примитивов,* с помощью которых пользователь или какая-либо активность получает доступ к сервису.

Модель TCP/IP
Рассмотрим другую эталонную модель, прототипом для которой послужил прародитель Интернета — сеть ARPA. С самого начала эта сеть задумывалась как объединение нескольких разных сетей. Одной из основных целей этого проекта была разработка унифицированных способов соединения сетей для создания систем передачи данных, обладающих высокой живучестью. Так появилась модель TCP/IP, получившая название по именам двух основных протоколов: протокола управления передачей — TCP (Transmission Control Protocol) и межсетевого протокола — IP (Internet Protocol). Связь в этой сети должна поддерживаться до тех пор, пока источник информации и получатель информации работоспособны. Архитектура сети ARPA не должна была ограничивать приложения, начиная от простой передачи файлов до передачи речи и изображения в реальном времени. Модель TCP/IP включает в себя 3 основных уровня:

- Межсетевой уровень
- Транспортный уровень
- Уровень приложений

В модели TCP/IP нет уровней сессии и представления, поскольку необходимость в них была неочевидна для ее создателей.
Модели TCP/IP и OSI имеют много общего. Обе эти модели имеют уровневую организацию и поддерживают понятие стека протоколов. Назначение их уровней примерно одинаковое. Все уровни этих моделей от транспортного и ниже используют протоколы для поддержки взаимодействия типа точка—точка, не зависящего от организации СПД, а все уровни выше транспортного ориентированы на приложения.
Наибольшее значение модели OSI методологическое: в ней явно определены и четко выделены понятия сервиса, интерфейса, протокола, уровня. В модели TCP/IP нет столь же четкого выделения этих понятий. В ней понятие протокола оторвано от остальных частей, и в ней нет единой, хорошо продуманной, концепции построения. Этот факт является следствием того, как создавались эти модели.

Билет № 10.
Примеры систем передачи данных (SMDS, SMDS/DQDB, Frame Relay, ISDN, B-ISDN, ATM) и их сравнение.

СПД на основе стандарта Bluetooth. Стандарт Bluetooth относится к классу беспроводных персональных сетей (WPAN). Спецификация Bluetooth представляет собой стандарт обмена информацией между такими устройствами, как карманные и обычные персональные компьютеры, мобильные телефоны, ноутбуки, принтеры, цифровые фотоаппараты, мышки, клавиатуры, джойстики и наушники, на надежной недорогой повсеместно доступной радиочастоте для ближней связи. Протокол Bluetooth обеспечивает сообщение между этими устройствами на расстоянии 10... 100 м друг от друга. Эта спецификация была разработана компанией Ericsson. Для связи в этом протоколе используется свободный от лицензирования диапазон частот 2,40...2,48 ГГц. Максимальная скорость передачи – не более 2 Мбит/с, максимальный размер пакета данных – 341 байт.
СПД на основе стандарта X.25. Стандарт X.25 в настоящее время используют некоторые телефонные сети, особенно в Европе, для подключения банкоматов. Этот стандарт, разработанный Международным союзом электросвязи в 1970-х гг., определяет интерфейс между СПД с коммутацией пакетов и терминалом, а также взаимодействие терминалов через сеть передачи данных с коммутацией пакетов. Рекомендации стандарта X.25 определяют способ передачи цифровых данных с коммутацией пакетов по телефонным каналам, который характеризуется следующими свойствами: пакеты длиной до 128 байт; доступная скорость 64 Кбит/с; ориентация на соединение и поддержку режима коммутируемых виртуальных каналов и режима постоянного виртуального канала. Поскольку к моменту появления стандарта X.25 в мире было уже много оконечных устройств, не рассчитанных на него, появилось решение этой проблемы в виде устройства PAD (Packet Assembler Disassembler), с помощью которого можно было подключать к данной сети оконечные устройства, не предназначенные для работы в ней.
СПД на основе Frame Relay. Ретрансляция кадров – это метод доставки сообщений в сетях передачи данных с коммутацией пакетов. Достоинства: малое время задержки сообщений, простой формат кадров, содержащих минимум управляющей информации, независимость от протоколов верхних уровней эталонной модели ISO. Службу FR можно рассматривать как аренду виртуального соединения, позволяющую передавать пакеты длиной до 1 600 байт. Служба FR предоставляет минимальный сервис. Если фрейм поступил с ошибкой, то он просто сбрасывается, и дело пользователя определить, какой фрейм пропущен, и как его восстановить. FR не поддерживает уведомления о доставке и обычное управление потоком. В настоящее время служба FR предоставляет услуги по передаче пакетов данных через постоянные виртуальные каналы (PVC) и интерфейс пользователь—сеть (UNT). Маршрутизация пакетов по PVC фиксируется заранее в момент установления такого канала и впоследствии не изменяется. В существующую версию FR не вошли коммутируемые виртуальные каналы (SVC) и интерфейс межсетевого взаимодействия.
Универсальные СПД и асинхронный способ передачи. Проблема интеграции СПД. Создание единой сети, обеспечивающей такую высокую скорость передачи, которая будет способна поддерживать любую услугу - unified communication. ATM (Asynchronous Transfer Mode). Главная идея - передача данных малыми порциями фиксированной длины (*ячейками).* Каждая ячейка имеет длину 53 байт (48 байт данные + 5 байт заголовок). Преимущества: ячейки удобно использовать для управления и передачи разнородных данных; при больших скоростях связи проще управлять коммутацией небольших ячеек, чем использовать старую технику мультиплексирования. ATM — это технология, ориентированная на соединение (виртуальное соединение). Доставка данных не гарантируется, но их порядок сохраняется. Скорость передачи данных от 155 до 622 Мбит/с.

Свойство	DQDB	Bluetooth	X.25	Frame Relay	ATM
Ориентированность на соединение	Есть	Нет	Есть	Есть	Есть
Стандартная скорость передачи, Мбит/с	45	2	0,064	1,5	155
Коммутируемость	Нет	Нет	Есть	Нет	Есть
Фиксированная длина кадра	Есть	Нет	Нет	Нет	Есть
Максимальная длина кадра, байт	44	341	128	1 600	53
Постоянные виртуальные каналы	Нет	Нет	Есть	Есть	Есть
Групповое вещание	Нет	Есть	Нет	Нет	Есть

Билет № 11.
Требования, предъявляемые к современным вычислительным сетям.

Главным требованием, предъявляемым к современным компьютерным сетям, является обеспечение пользователям доступа к вычислительным сервисам. Все остальные требования характеризуют качество реализации этих сервисов.
Производительность характеризует скорость работы сети. Эта характеристика определяется числом услуг, предоставляемых сетью в единицу времени. Под услугой может пониматься пропускная способность — число пакетов, пройденных через сеть за секунду, минуту, час, день, причем различают среднюю, мгновенную, пиковую, минимальную пропускную способность сети. Это также может быть время выполнения определенной операции.
Индекс, характеризующий только работу СПД, называется *временем передачи* — это время от поступления пакета данных на вход СПД до появления его на выходе. *Надежность* характеризует способность сети выполнять операции, и если операция запущена, то всегда ли она корректно завершится. Имеется несколько подходов к определению этой характеристики:

- через измерение надежности устройств (времени наработки на отказ, вероятности отказа, интенсивности отказов);
- коэффициента готовности — доли времени, в течение которого система может быть использована;
- вероятности доставки пакета через транспортное соединение;
- вероятности искажения пакета в транспортном соединении;
- отказоустойчивости — способности обнаруживать ошибки функционирования и устранять их.

Безопасность характеризует защищенность информации и ресурсов сети от несанкционированного использования и изменения:

- транспортной среды;
- СПД;
- вычислительных ресурсов;
- данных, программ (доступа, изменения).

Конфиденциальность данных характеризует возможность обеспечения доступа к данным лишь тем, кто имеет на это право.
Целостность данных определяет возможность изменения данных только теми, кто имеет на это право.
Расширяемость характеризует сложность изменения конфигурации сети. Например, если для подключения новой машины в сеть необходимо остановить работу СПД, то такую сеть вряд ли можно назвать легко расширяемой.
Масштабируемость характеризует способность сети плавно увеличивать вычислительную мощность без деградации ее производительности в целом.
Прозрачность характеризует, насколько ≪просто≫ пользоваться сетью. Чем сложнее доступ пользователю к требуемому сервису в сети, тем менее она прозрачна.

- сеть сама распределяет ресурсы и управляет ими;
- обеспечивает среду для разработки и выполнения программ;
- является поставщиком разнообразного сервиса;
- для пользователя она прозрачна (он ее не видит), он лишь сообщает что ему требуется, а как и где это взять решает сеть.

Важной характеристикой сети является способность по одним и тем же СПД, т.е. по так называемым унифицированным системам передачи, передавать разнородные потоки данных.
Управляемость характеризует возможность управлять и контролировать работу каждого отдельного устройства в сети из единого центра.
Совместимость характеризует способность подключать оборудование и программное обеспечение разных производителей.

Билет № 12.
Что такое стандарт на взаимодействие в сетях, кто, как и для чего вводит стандарты?

Стандартизация — это деятельность по установлению норм, правил и характеристик в целях обеспечения:
• безопасности продукции, работ и услуг для окружающей среды, жизни, здоровья и имущества человека;
• технической и информационной совместимости, а также взаимозаменяемости продукции;
• качества продукции, работ и услуг в соответствии с уровнем развития науки, техники и технологии;
• единства измерений;
• экономии всех видов ресурсов;
• безопасности хозяйственных объектов с учетом риска возникновения природных и техногенных катастроф и других чрезвычайных ситуаций;
• обороноспособности и мобилизационной готовности страны.
Стандарт — документ, устанавливающий комплекс норм, правил и требований к объекту стандартизации.

Классификация организаций (по масштабу):
• международные организации;
• региональные международные организации;
• международные промышленные консорциумы и профессиональные ассоциации;
• национальные организации;
• отраслевые организации (действующие в рамках отрасли отдельного государства).

Официальные организациям в международной системе стандартизации информационных технологий:
ISO (International Organization for Standardization) — Международная организация по стандартизации.
IEC (International Electrotechnical Commision) — Международная электротехническая комиссия — МЭК;
ITU (International Telecommunication Union) — Международный союз электросвязи — МСЭ.

Международная организация по стандартизации - ISO - была создана в 1946 г. Деятельность ИСО касается стандартизации во всех областях, кроме электротехники, электроники и связи. Сейчас в состав ИСО входят 135 стран в лице своих национальных организаций по стандартизации. Непосредственную работу по созданию международных стандартов выполняют технические комитеты (ТК), подкомитеты (ПК), которые могут учреждать ТК, и рабочие группы (РГ) по конкретным направлениям деятельности. Обеспечивают работу всех секретариатов ТК и ПК 35 организаций — членов ИСО. Официальные языки ИСО — английский, французский, русский.
Схема разработки международного стандарта следующая: заинтересованная сторона в лице организации — члена ИСО, технического комитета или комитета Генеральной ассамблеи (либо организации, не являющейся членом ИСО) направляет в ИСО заявку на разработку стандарта. Генеральный секретарь по согласованию с организациями — членами ИСО предоставляет в Техническое руководящее бюро предложение о создании соответствующего ТК. Этот ТК создается при следующих условиях: если большинство организаций — членов ИСО голосуют «за» и не менее пяти из них намерены стать его участниками, а Техническое руководящее бюро убеждено в международной значимости будущего стандарта. Все вопросы в процессе работы обычно решаются на основе консенсуса организаций–членов ИСО, активно участвующих в деятельности ТК. После достижения консенсуса ТК передает проект стандарта в Центральный секретариат для регистрации и рассылки всем комитетам на голосование. Если проект одобряется 75 % голосовавших, он публикуется в качестве международного стандарта.
Стандарты ИСО не имеют статуса обязательных для всех стран — участниц этой организации. Решение вопроса о применении стандарта ИСО связано в основном со степенью участия страны в международном разделении труда и состоянием ее внешней торговли. В российской системе стандартизации нашли применение около половины всех стандартов ИСО.

Международная электротехническая комиссия IEC —международная организация, занимающаяся разработкой стандартов в области электротехники. Образована в 1906 г., является добровольной неправительственной организацией. Ее деятельность связана со стандартизацией физических характеристик электротехнического и электронного оборудования. Основное внимание МЭК уделяет таким вопросам, как электроизмерения, тестирование, способы использования и безопасность электротехнического и электронного оборудования. Членами МЭК являются национальные организации (комитеты) стандартизации технологий в соответствующих отраслях, представляющие интересы своих стран в деле международной стандартизации. В настоящее время в состав МЭК входит более 50 стран. С организационной точки зрения МЭК похожа на ИСО.

Международный союз электросвязи – ITU (МСЭ) - международная межправительственная организация, занимающаяся вопросами стандартизации и развития электросвязи. МСЭ объединяет более 500 правительственных и неправительственных организаций. Основная задача МСЭ состоит в координации разработки сбалансированных на международном уровне правил и рекомендаций, предназначенных для построения и использования глобальных систем передачи данных и их сервисов. МСЭ —старейшая международная профессиональная организация. Она была основана в 1865 г. после подписания 20 европейскими государствами первой международной конвенции по телеграфии.
Поскольку деятельность ИСО и деятельность МЭК часто пересекались в области стандартизации информационных технологий (ИТ), то в 1987 г. этими организациями было принято совместное решение о создании Объединенного технического комитета 1 (Joint Technical Committee 1 —JTCI), основная функция которого была определена как формирование системы стандартов в области ИТ и их расширений для конкретных сфер деятельности. В документах, регламентирующих работу JTC1, определено, что *информационные технологии включают в себя спецификацию, проектирование и разработку систем и средств, имеющих дело со сбором, представлением, обработкой, безопасностью, передачей, организацией, хранением и поиском информации, а также ее обменом и управлением*. Основными задачами JTC1 являются разработка, поддержание, продвижение стандартов ИТ, необходимых для глобального рынка, удовлетворяющих требованиям бизнеса и пользователей и относящихся: к проектированию и разработке систем и средств ИТ; производительности и качеству продуктов и систем ИТ; безопасности систем ИТ и информации; переносимости прикладных программ; интероперабельности продуктов и систем ИТ; унифицированным средствам и окружениям; гармонизированному словарю понятий в области ИТ; дружеским и эргономичным пользовательским интерфейсам.

Региональные межправительственные организации по стандартизации:
• общеевропейские организации по стандартизапии CEN, CENELEC, ETSI;
• межскандинавская организация по стандартизации (INSTA);
• международная ассоциация стран Юго-Восточной Азии (АСЕАН);
• панамериканский комитет стандартов (КОПАНТ);
• межгосударственный совет стран — членов СНГ (МТС).

CEN (the European Committee for Standardization) —европейский комитет по стандартизации широкого спектра товаров, услуг и технологий, в том числе связанных с областью ИТ. Существует с 1961 г. Членами CEN являются национальные организации по стандартизации европейских государств: Австрии, Бельгии, Великобритании, Греции, Дании, Германии, Испании, Исландии, Италии, Люксембурга, Норвегии, Нидерландов, Португалии, Финляндии, ФРГ, Франции, Швеции, Швейцарии. Это закрытая организация, в которую до 1992 г. входили только члены ЕС и ЕАСТ (Европейской Ассоциации Свободной Торговли). Высший орган CEN —Генеральная ассамблея, где представлены национальные организации по стандартизации, правительственные органы стран —членов ЕС и ЕАСТ, а также ассоциированные организации.
В 1998 г. было создано новое подразделение CEN, названное TSSS (the Information Society Standardization System), целью которого является обеспечение участников рынка европейского информационного сообщества системой стандартов для продуктов и сервисов в области информационных и телекоммуникационных технологий. Основная цель CEN —содействие развитию торговли товарами и услугами посредством: разработки европейских стандартов (евронорм, EN); применения в странах —членах этого комитета стандартов ИСО и МЭК; сотрудничества со всеми организациями региона, занимающимися стандартизацией; предоставления услуг по сертификации на соответствие европейским стандартам.

ETS1 (European Telecommunications Standards Institute) образован в 1988 г. Основной задачей является разработка стандартов в области сетевой инфраструктуры. ETSI ведет работы по следующим основным направлениям: кабельные сети; беспроводные и мобильные сети; прикладные телекоммуникационные сервисы глобальной информационной инфраструктуры; архитектура сетей и управление сетями; межотраслевые решения, включая решения по электромагнитной совместимости, терминальному оборудованию, эргономике и человеческому фактору. Стандарт для мобильной связи GSM. ETSI действует в тесном сотрудничестве с CEN.

Стандартизация в СНГ
Соглашение о проведении согласованной политики в области стандартизации, метрологии и сертификации (1992). Межгосударственный совет стран-участниц СНГ (МГС), в котором представлены все национальные

организации по стандартизации этих государств. МГС принимает межгосударственные стандарты. Работа по стандартизации ведется в соответствии с программами, которые МГС составляет на основе обобщения предложений, поступающих от национальных органов по стандартизации. МГС подписал соглашения с МЭК и CEN о сотрудничестве. В МГС также принято соглашение об условиях прямого применения европейских стандартов в качестве межгосударственных для стран СНГ.

***Промышленные консорциумы и профессиональные ассоциации*.**
Заинтересованность участников консорциума в достижении конечного результата в сжатые сроки => высокая скорость процесса разработки и согласования стандартов, успешно решаются вопросы, связанные с финансовым обеспечением проектов стандартизации.
Отслеживает и специфицирует деятельность консорциумов CEN/ISSS. К началу 2000 г. этой организацией было зарегистрировано около 150 консорциумов, работающих в области стандартизации ИТ. Наиболее известными представителями этой группы организаций–разработчиков стандартов являются:
IEEE (Institute of Electrical and Electronic Engineers —Институт инженеров по электротехнике и электронике) —профессиональная международная организация, являющаяся разработчиком ряда важных международных стандартов ИТ;
OMG (Object Management Group —группа управления объектами) — международный консорциум, осуществляющий разработку стандартов унифицированного распределенного программного обеспечения, созданного на принципах объектно-ориентированного подхода;
ECMA (European Computer Manufacturers Association —Европейская ассоциация производителей вычислительных машин) — международная ассоциация, осуществляющая промышленную стандартизацию информационных и коммуникационных систем;
W3C (World Wide Web Consortium) —консорциум, который специализируется в области разработки и развитии стандартов WWW-технологий, например HTTP, HTML, URL, XML;
ATM Forum (Asynchronous Transfere Mode) —консорциум, осуществляющий разработку и развитие стандартов широкополосных сетей асинхронного режима передачи данных;
Gigabit Ethernet Alliance —консорциум, осуществляющий разработку стандартов технологий Ethernet нового поколения (совместно с комитетом IEEE с индексом 802.3z), обеспечивающих скорость передачи данных в 1 Гбит/с.
Следует подчеркнуть, что одной из главных тенденций процесса стандартизации является все более тесная интеграция деятельности этих организаций, направленная на создание единой системы стандартизации информационного общества.

Национальные организации по стандартизации. В каждой современной индустриально развитой стране существует одна организация по стандартизации, представляющая данную страну в ИСО в качестве участника международного процесса стандартизации. Задачи: участвуют в разработке и принятии международных стандартов с учетом национальных интересов; выполняют локализацию и адаптацию международных стандартов для их успешного применения в своих странах, способствуют разработке национальных стандартов в соответствии с международными стандартами; передают в ИСО для стандартизации на международном уровне разработанные ими спецификации, являющиеся национальными стандартами. Примерами национальных организаций являются:
ANSI (American National Standards Institute) —американский институт национальных стандартов.
AFNOR (Association Francaise de Normalisation) —французская ассоциация по стандартизации.
BSI (British Standards Institute) —британский институт стандартов;
DIN (Deutsches Institute fur Normung e.v.) —германская организация национальных стандартов;
JISC (Japanese Industrial Standards Committee) —японский комитет промышленных стандартов.
Национальным органом по стандартизации в России является Госстандарт России.

Билет № 13.

Теоретические основы передачи данных (ограничения на пропускную способность передачи сигналов, взаимосвязь пропускной способности канала и ширины его полосы пропускания). Среды передачи (магнитные носители, витая пара, среднеполосный и широкополосный кабели, оптоволокно, сравнение кабелей и оптоволокна).

Любая информация может передаваться с помощью электромагнитных импульсов (сигналов). В зависимости от среды передачи и организации СПД используются либо аналоговые, либо цифровые сигналы.
Любой сигнал можно рассматривать либо как функцию времени, либо как функцию частоты (как композицию составляющих сигналов - гармоник). Важной характеристикой сигнала является *ширина его полосы*, которая покрывает весь спектр частот гармоник, составляющих сигнал. Чем шире эта полоса, тем больше информационная емкость сигнала. При создании любой СПД приходится искать компромисс между четырьмя основными факторами: шириной полосы сигнала, скоростью передачи сигналов, уровнем шумов и искажений сигнала, допустимым уровнем ошибок при передаче.
Частотное представление функции основано на том факте, что любая функция от вещественной переменной может быть представлена в виде ряда Фурье
Характеристику канала, определяющую спектр частот, которые физическая среда, из которой сделана линия связи, образующая канал, пропускает без существенного понижения мощности сигнала, называют полосой пропускания.
Данные – это то, с помощью чего мы описываем явление или объект.
Сигнал – это представление данных.
Передача – это процесс взаимодействия передатчика и приемника с целью получения приемником сигналов от передатчика.
Сигналы могут иметь непрерывную или дискретную форму. В первом случае говорят об аналоговом сигнале, во втором - о цифровом.
Большое значение имеет количество уровней, которое может иметь сигнал. Чем больше число уровней сигнала, тем больше информации можно передать за один переход с уровня на уровень.
Процесс передачи также может иметь аналоговую или цифровую формы. *Аналоговая* передача предполагает непрерывное изменение параметров передачи. *Цифровая* –резкое, дискретное изменение параметров передаваемого сигнала или импульса.
Максимальную скорость, с которой канал способен передавать данные, называют *пропускной способностью канала* или *битовой скоростью.*
Теорема Найквиста (взаимосвязь между пропускной способности канала и шириной его полосы пропускания): $V_{\text{max data rate}} = 2H \log_2 M$ *бит/сек*, где $V_{\text{max data rate}}$ – максимальная скорость передачи H – ширина полосы пропускания канала, выраженная в Гц, M - количество уровней сигнала.
Шум в канале: отношение мощности полезного сигнала к мощности шума: S/N. Измеряется в децибелах: $10\log_{10}(S/N)$dB.
Теорема Шеннона: максимальная скорость передачи данных по каналу с шумом равна $H \log_2 (1+S/N)$ *бит/сек*. где S/N - соотношение сигнал-шум в канале.

Сигнальная скорость, или скорость модуляции - скорость изменения значения сигнала. Ихмеряется в ***бод***ах. Если скорость изменения значения сигнала b бод, то это не означает, что данные передается со скоростью b бит/сек. Многое зависит способа кодирования сигнала: одно изменение значения может кодировать сразу несколько бит.

Среды передачи

Назначение физического уровня - передать данные в виде потока бит от одной машины к другой. Для передачи можно использовать разные физические среды. Каждая из сред имеет свои уникальные характеристики, такие как:

- полоса пропускания
- пропускная способность
- задержка
- стоимость
- простота прокладки
- сложность в обслуживании.

Кроме них важно учитывать также такие характеристики как, например, достоверность передачи, затухание, помехоустойчивость и т.д.

Магнитные носители

Магнитная лента или магнитный диск в сочетании с обычным транспортным средством могут быть прекрасной физической средой передачи данных. Это так особенно там, где высокая пропускная способность и низкая стоимость передачи в расчете на один бит – ключевые факторы.

Витая пара

Для многих приложений нужен оперативный обмен информацией. Самой старой и все еще используемой средой передачи является **витая пара**. Витая пара состоит из двух медных изолированных проводов, один из которых обвит вокруг другого. Вьющийся провод предназначен для устранения взаимного влияния между соседними витыми парами. Витая пара широко используется в телефонии. Особенно между абонентами и местной АТС, линии из витой пары могут иметь протяженность до нескольких километров без промежуточного усиления. Витые пары объединяются в многопарные кабели. Витая пара может быть использована для передачи как цифрового, так и аналогового сигналов. Пропускная способность зависит от толщины линий и расстояния. Скорость в несколько мегабит в секунду вполне достижима с помощью соответствующих методов передачи. На коротких расстояниях была достигнута скорость до 1 Г бит/сек. На больших расстояниях скорость передачи не превышает 4 М бит/сек.
Кабели категории 3 содержат по четыре витые пары с невысокой плотностью навивки, и имеют полосу пропускания до 16 МГц. Кабель категории 5 имеет тоже четыре пары, но с более плотной навивкой, что позволяет достичь более высоких скоростей, и имеют полосу пропускания 100 МГц.

Коаксиальные кабели

Подобно витой паре у коаксиального кабеля есть два проводника. Центральный проводник представляет собой медный проводник, окруженный изолятором. Эта конструкция помещается внутри второго цилиндрического проводника, который обычно представляет собой сплетенную плотную металлическую сетку. Все это закрывается плотным защитным слоем пластика. Обычно толщина коаксиала от 1 до 2.5 см. У коаксиала полоса пропускания шире и характеристики по затуханию сигнала лучше, чем у витой пары. Поэтому эти кабели применяют на больших расстояниях и по ним могут передавать одновременно несколько потоков данных от разных компьютеров. Коаксиальные кабели используют для передачи как аналоговых, так и цифровых сигналов. Основными ограничениями скорости и расстояния передачи без усиления являются в этих кабелях затухание сигнала, тепловой шум и интермодуляционный шум. Последний вид шума возникает когда всю полосу пропускания кабеля разбивают на более узкие полосы и каждую такую полосу используют как отдельный канал. Есть два основных вида коаксиальных кабелей: узкополосный с волновым сопротивлением 50 Ом и широкополосный с волновым сопротивлением 75 Ом. Узкополосный кабель позволяет достигать скорости в несколько Гбит/сек, при длине в 1-2 км при высокой помехозащищенности. Второй вид коаксиальных кабелей используют в телевидении и называют высокочастотным кабелем
В двух кабельных системах прокладывается сразу два кабеля: один кабель используется для входящего потока, а второй для исходящего

Оптоволокно

Для использования оптической связи нужен источник света, светопроводящая среда, детектор, преобразующий световой поток в электрический. На одном передающем конце волоконнооптической линии находится источник света, световой импульс от этого источника проходит по тонкому светопроводящему волокну и попадает на детектор, который преобразует этот импульс в электрический.
Одна из основных проблем создания оптоволоконных систем состояла в том, чтобы не дать световому пучку рассеяться через боковую поверхность силиконового шнура. Количество рассеиваемой энергии зависело от угла падения светового луча на стенки шнура.
При углах больше некоторого критического угла, называемого углом полного внутреннего отражения вся энергия луча отражается обратно внутрь.
Если сделать силиконовый шнур толщиной близкой к длине волны источника света, то этот шнур будет работать, как провод для тока, без потерь на внутреннее отражение. По такому **одномодовому** шнуру можно передавать со скоростью в несколько Гбит/сек на сотню километров без промежуточного усиления.
Поскольку можно испускать несколько лучей так, чтобы они попадали на границы шнура под углом большим угла полного внутреннего отражения, то по одному шнуру можно пускать несколько лучей. Каждый луч, как говорят, имеет свою моду. Так мы получаем **многомодовый** шнур.
Оптоволокно делают из стеклоподобного материала, которое в свою очередь делают из песка и других широко распространенных материалов.

Затухание оптического сигнала в стекле зависит от длины волны источника света. Затухание измеряется в dB по формуле $10\log_{10}\frac{Tp}{Rp}$, где Tp – мощность передаваемого сигнала, Rp – мощность полученного сигнала.

Для передачи используются три полосы с длинами волн 0.85, 1,30 и 1.55 мкм. Две последние обладают тем замечательным свойством, что их затухание составляет менее 5% на километр. Длина волны в 0.85 мкм имеет большее затухание, но хороша тем, что лучше соответствует возможностям лазерных источников света. У всех трех полос ширина полосы пропускания от 25 000 до 30 000 ГГц.

Другую проблему при использовании оптоволокна представляет дисперсия: исходный световой импульс по мере распространения теряет начальную форму и размеры. Величина этих искажений также зависит от длины волны. Одно из возможных решений - увеличить расстояние между соседними сигналами. Однако это сократит скорость передачи. К счастью, исследования показали, что если придать сигналу некоторую специальную форму, то дисперсионные эффекты почти исчезают и сигнал можно передавать на тысячи километров. Сигналы в этой специальной форме называются **силитонами.**

В заключение будет полезно сравнить возможности медного кабеля и оптоволокна:

1. Ширина полосы пропускания у оптоволокна несравненно больше, чем у медного кабеля, что позволяет достичь скорости в сотни Гбит/сек на расстояниях в десятки километров.
2. Оптоволокно компактнее и меньше весит. При той же пропускной способности коаксиальный кабель и кабель из витых пар существенно тяжелее оптоволокна. Это существенный фактор, влияющий на стоимость и требования к опорным конструкциям. Например, 1 км 1000 парника весит 8 тонн, а оптоволокно аналогичной пропускной способности – 100 кг.
3. Затухание сигнала в оптоволокне существенно меньше, чем в коаксиале и витой паре, и остается постоянным для широкого диапазона частот.
4. Оптоволокно не восприимчиво к внешним электромагнитным излучениям. Поэтому ему не страшны интерференция, импульсные шумы и взаимные наводки. Оптоволокно не излучает энергию. Поэтому не влияет на работу другого оборудования. Его трудно обнаружить, следовательно найти и повредить.
5. Чем меньше репитеров, тем дешевле система и меньше источников ошибок. С этой точки зрения оптоволоконные системы достигли большего совершенства. Для этих систем среднее расстояние между репитерами – сотни километров. Для коаксиала или витой пары тот же показатель равен нескольким километрам.

Билет № 14.

Теоретические основы передачи данных (ограничения на пропускную способность передачи сигналов, взаимосвязь пропускной способности канала и ширины его полосы пропускания). Передача цифровых данных цифровыми сигналами.

Цифровой сигнал – это дискретная последовательность импульсов по напряжению, каждый из которых имеет ступенчатую форму. Каждый импульс – это единичный сигнал. В общем случае, данные в двоичной форме при передаче кодируются так, что один бит данных отображается в несколько единичных сигналов. В простейшем случае это соответствие имеет однозначный характер: один бит – один сигнал.
Если все единичные сигналы имеют одинаковую полярность, то говорят, что сигнал *униполярный*.
Скорость передачи данных – это количество бит в секунду, которые передают с помощью сигналов. Эту скорость также называют *битовой скоростью*. *Продолжительность* или *длина бита* – это интервал времени, который нужно передатчику, чтобы испустить надлежащий единичный сигнал. При скорости передачи данных *R* бит/сек, длина бита равна *1/R*.
Прежде всего, приемник должен быть строго настроен на длину бита. Он должен уметь распознавать начало и конец передачи каждого бита. Уметь распознавать уровень сигнала: низкий или высокий. Например, эти задачи решаются измерением уровня сигнала в середине длины бита и сравнением результата измерения с пороговым значением. Из-за шума на линии при этом могут возникать ошибки.
Есть три важных фактора влияющие на правильность передачи: уровень шума, скорость передачи данных и ширина полосы пропускания канала. Есть еще один фактор, влияющий на передачу данных: это способ представления (кодировки) данных на физическом уровне.

Основными критериями сравнения различных способов кодирования данных на физическом уровне являются:

- Ширина спектра сигнала: Чем меньше высокочастотных составляющих в сигнале, тем уже ширина полосы пропускания может быть при передаче. Важным также является отсутствие постоянной составляющей (приводит к наличию постоянного тока между приемником и передатчиком, что крайне нежелательно). Чем шире спектр, тем сильнее искажения.
- Синхронизация между приемником и передатчиком: приемник должен точно определять начало и конец битового интервала. На небольших расстояниях можно использовать дополнительную линию синхронизации - тактирующая схема (часы) выдает строго через определенные промежутки синхроимпульсы. Другое решение этой проблемы состоит в создании самосинхронизирующихся кодов.
- Обнаружение ошибок.
- Чувствительность к шуму: За счет надлежащих ухищрений в схеме кодировки данных можно добиться очень высокой производительности при передаче даже при наличии очень высокого уровня шума.
- Стоимость и скорость.

Потенциальный код NRZ 0 – высокий потенциал 1 – низкий потенциал
Биполярный код NRZI 0 – нет перепада уровня сигнала в начале битного интервала 1 – перепад уровня сигнала в начале интервала
Биполярный код AMI 0 – отсутствие сигнала 1 – положительный или отрицательный потенциал, обратный по отношению к потенциалу в предыдущий период
Манчестерский код 0 – переход с высокого на низкий потенциал в середине интервала 1 – переход с низкого на высокий потенциал в середине интервала
Потенциальный код 2B1Q Использует 4 уровня сигналов, значение уровня определяется значением пары битов данных

Все схемы кодирования делятся на *потенциальные* и *импульсные*. У потенциальных кодов значение бита передается удержанием потенциала сигнала на определенном уровне в течении битового интервала. У импульсных кодов это значение передается перепадом (фронтом). Направление перепада с низкого на высокий или с высокого на низкий соответствует конкретному значению бита.

Потенциальный NRZ код

Основным недостатком этого кода является отсутствие синхронизации. На длинных последовательностях нулей или единиц потенциал на линии не меняется, и может произойти рассинхронизация между приемником и передатчиком, что приведет к ошибкам. Если исключить возможность появления длинных последовательностей 0 или 1 (например, использовать специальные устройства *скремблеры*), то этот метод может быть весьма эффективен.

Модификацией NRZ кода и хорошим примером дифференциального кодирования является NRZ-I код. Идея дифференциальных кодов состоит в том, чтобы кодировать не абсолютное значение текущего бита, а разницу значений между предыдущим битом и текущим. В случае NRZ-I кода если текущий бит – 0, то он кодируется тем же потенциалом, что и предыдущий бит, если текущий бит – 1, то он кодируется другим потенциалом, чем предыдущий. Основным достоинством NRZ-I кода по отношению NRZ коду является большая устойчивость к шуму.

Биполярный код AMI

У этого метода есть несколько существенных преимуществ по сравнению с NRZ кодами. Во-первых, в случае длительной последовательности 1 рассинхронизации не происходит. Каждая единица сопровождается изменение потенциала устойчиво распознаваемом приемником. Поскольку каждая единица сопровождается изменением потенциала, то не возникнет постоянной составляющей. Однако длинная последовательность 0 остается проблемой, и требуются дополнительные усилия, которые позволили бы избежать ее появления. Во-вторых, спектр сигнала здесь уже, чем у NRZ кодов. И, наконец, четко определенное правило чередования уровней позволяет обнаруживать единичные ошибки.

Биполярные импульсные коды

В *Манчестерском коде* данные кодируются фронтами в середине битового интервала. Этим достигаются две цели: синхронизация приемника и передатчика и передача данных: фронт перехода от низкого потенциала к высокому соответствует 1, а фронт перехода от высокого потенциала к низкому – 0.

В *дифференциальном Манчестерском коде* сигнал может менять свой уровень дважды в течении битового интервала. В середине интервала обязательно происходит изменение уровня. Этот перепад используется для синхронизации. При передаче 0 в начале битового интервала, происходит перепад уровней, при 1 – такой перепад отсутствует.

Преимущества биполярных импульсных методов:

- самосинхронизация
- отсутствие постоянной составляющей
- отсутствие единичных ошибок.

Потенциальный код 2B1Q

В этом методе каждые два последовательных бита (2B) передаются за один битовый интервал сигнала, который может иметь четыре состояния (1Q). Паре 00 соответствует потенциал -2.5 В, 01 соответствует -0.833 В, 11 – + 0.833 В, 10 – +2.5 В. У этого метода сигнальная скорость в два раза ниже, чем NRZ и AMI кодов, а спектр сигнала в два раза уже. Поэтому с помощью 2B1Q кода можно по одной и той же линии предавать данные в два раза быстрее. Однако, реализация этого метода требует более мощного передатчика и более сложного приемника, который должен различать не два уровня, а четыре.

В общем случае соотношение между битовой и сигнальной скоростью определяется формулой $D = \frac{R}{b}$,

где D – сигнальная скорость, R – битовая скорость в бит/сек., b – количество бит на единичный сигнал.

Билет № 15.

Теоретические основы передачи данных (ограничения на пропускную способность передачи сигналов, взаимосвязь пропускной способности канала и ширины его полосы пропускания). Передача аналоговых данных цифровыми сигналами.

Преобразование аналоговых данных в цифровой сигнал можно представить как преобразование аналоговых данных в цифровую форму. Этот процесс называют оцифровкой данных. Выполнив его, мы можем передать цифровые данные цифровым или аналоговым сигналом. Устройство АЦП (Аналогово-Цифровой Преобразователь) превращает аналоговые данные в цифровую форму, а устройство ЦАП (Цифро-Аналоговый преобразователь) выполняет обратную процедуру. Устройство, объединяющее в себе функции и АЦП и ЦАП, называют кодеком (кодер-декодер).

Импульсно кодовая модуляция (ИКМ) основана на следствии из теоремы Найквиста, которое утверждает если изменять параметры сигнала *f(t)* через регулярные интервалы времени с частотой не меньше, чем удвоенная частота самой высокочастотной составляющей сигнала, то полученная серия измерений будет содержать всю информацию об исходном сигнале и этот сигнал может быть восстановлен. Другой альтернативой ИКМ является метод Дельта модуляции. На исходную непрерывную функцию, представляющую аналоговый сигнал, накладывают ступенчатую функцию. Значения этой ступенчатой функции меняются на δ на каждом шаге квантования по времени T_s. Замена исходной функции на эту дискретную, ступенчатую функцию интересно тем, что поведение последней носит двоичный характер. На каждом шаге значение ступенчатой функции либо увеличивается на δ, будем представлять этот случай 1, либо сокращается на δ – случай 0.

Процесс передачи в случае Дельта модуляции организован следующим образом. В момент очередного замера текущее значение исходной функции сравнивается со значением ступенчатой функции на предыдущем шаге. Если значение исходной функции больше, придается 1, в противном случае – 0. Таким образом, ступенчатая функция всегда меняет свое значение.
У метода Дельта модуляции есть два параметра: величина шага δ и частота замеров или шаг квантования. Выбор шага δ – это баланс между ошибкой квантования и ошибкой перегрузки по крутизне
Когда исходный сигнал изменяется достаточно медленно, то возникает только ошибка квантования, чем больше δ, тем больше эта ошибка. Если же сигнал изменяется резко, то рост ступенчатой функции может отставать. Это вид ошибки растет с уменьшением δ.
Положение можно улучшить, увеличив частоту замеров, но это увеличит битовую скорость на линии.

Билет № 16.

Теоретические основы передачи данных (ограничения на пропускную способность передачи сигналов, взаимосвязь пропускной способности канала и ширины его полосы пропускания). Передача цифровых данных аналоговыми сигналами.

Примером передачи данных в цифровой форме с помощью аналоговых сигналов является использование телефонных сетей для передачи цифровых данных. Телефонные сети были созданы для передачи и коммутации аналоговых сигналов в голосовом диапазоне частот от 300 до 3400 Гц.
Модем(МОдулятор–ДЕМодулятор) – прибор преобразует цифровой сигнал в аналоговый и наоборот в надлежащем диапазоне частот.
Аналоговая модуляция заключается в преобразовании одного или нескольких параметров из трех основных параметров несущего сигнала: амплитуды, частоты и фазы.
Есть три основных метода модуляции для преобразования цифровых данных в аналоговую форму:

- амплитудная модуляция
- частотная модуляция
- фазовая модуляция.

В случае амплитудной модуляции двоичные 0 и 1 представлены аналоговым сигналом на частоте несущей, но разной амплитуды. Обычно 0 соответствует сигнал с нулевой амплитудой. Таким образом, при амплитудной модуляции сигнал *S(t)* имеет вид

$$S(t) = \begin{cases} A\cos(2\pi f_c t) & \textit{двоичная} - 1 \\ 0 & \textit{двоичный} - 0 \end{cases},$$

где $A\cos(2\pi f_c t)$ несущий сигнал с амплитудой *A*. Метод амплитудной модуляции не очень эффективен по сравнению с другими методами, т.к. он очень чувствителен к шумам.

При частотной модуляции двоичные 0 и 1 представляют сигналами разной частоты, сдвинутой, как правило, по отношению к частоте несущей на одинаковую величину, но в противоположном направлении:

$$S(t) = \begin{cases} A\cos(2\pi f_1 t) & \textit{для} - 1 \\ A\cos(2\pi f_2 t) & \textit{для} - 0 \end{cases},$$ где $f_c=f_1-\Delta=f_2+\Delta$ где Δ – сдвиг по частоте.

Частотную модуляцию чаще всего применяют в радиомодемах на частотах от 3 МГц до 30 МГц, а также в высокочастотных кабелях локальных сетей.

Фазовая модуляция состоит в представлении цифровых данных сдвигом фазы несущего сигнала. Для дифференциальной фазовой модуляции получаем

$$S(t) = \begin{cases} A\cos(2\pi f_c t + \pi) & \textit{для} - 1 \\ A\cos(2\pi f_c t) & \textit{для} - 0 \end{cases},$$

Эффективность использования полосы пропускания можно существенно повысить, если единичный сигнал будет кодировать несколько бит. В общем случае скорость модуляции $D = \frac{R}{b} = \frac{R}{\log_2 L}$, где *D* – скорость модуляции (сигнальная скорость), *R* – битовая скорость (скорость передачи данных), *L* – число разных уровней единичных сигналов, *b* – число бит на единичный сигнал.

Билет № 17.

Теоретические основы передачи данных (ограничения на пропускную способность передачи сигналов, взаимосвязь пропускной способности канала и ширины его полосы пропускания). Передача аналоговых данных аналоговыми сигналами.

Потребность возникает при использовании радио каналов. Модуляция, т.е. объединение исходного сигнала *m(t)* и несущей частоты f_c , позволяет нужным образом изменять параметры исходного сигнала и, тем самым, упростить решение ряда технических проблем. Кроме этого, модуляция позволяет использовать методы мультиплексирования.

Три способа модуляции амплитудная модуляция, частотная и фазовая.

При амплитудной модуляции форма результирующего сигнала определяется формулой:
$S(t) = [1 + n_a x(t)] \cos 2\pi f_c t$,
где f_c – частота несущей, n_a – индекс модуляции, который определяют как отношение амплитуды исходного сигнала к амплитуде несущего сигнала.
В наших обозначениях $m(t)=1+n_a x(t)$.

Форма результирующего сигнала при частотной модуляции определяется следующим выражением:
$S(t) = A_c \cos(2\pi f_c t + n_f m(t))$, где n_f – индекс частотной модуляции.

Сигнал, получаемый фазовой модуляцией, определяет соотношение:
$S(t) = A_c \cos(2\pi f_c t + n_p m(t))$,где n_p – индекс частотной модуляции.

Хотя все эти три вида модуляции порождают сигнал *S(t),* спектр которого симметричен относительно f_c , но в случае амплитудной модуляции он проще по составу. В случае частотной и фазовой модуляций требуется, в общем случае, более широкая полоса пропускания.
Широко распространенным случаем аналоговой модуляции является метод квадратичной амплитудной модуляции QAM. Именно этот метод используется в асимметричных цифровых линиях – ADSL. Метод QAM – это комбинация амплитудной и фазовой модуляций. Идея этого метода состоит в том, что можно по одной и той же линии послать одновременно два разных сигнала с одинаковой несущей частотой, но сдвинутых по фазе друг относительно друга на 90º. Каждый сигнал генерируется методом амплитудной модуляции.

Билет № 18.

Физические среды передачи данных. Беспроводная связь (электромагнитный спектр, радиопередача, микроволновая передача, видимое излучение).

Назначение физического уровня — передавать данные в виде потока битов от одной машины к другой. При этом для передачи данных можно использовать различные физические среды, каждую из которых характеризуют следующие параметры:

- ширина полосы пропускания;
- пропускная способность;
- задержка сигнала;
- стоимость;
- сложность прокладки;
- сложность обслуживания.

Кроме перечисленных, физическую среду характеризуют и другие параметры, например:

- достоверность передачи;
- затухание;
- помехоустойчивость.

Магнитные носители.

Магнитная лента или магнитный диск в сочетании с обычным транспортным средством (автомашиной, железной дорогой и т.п.) могут быть прекрасной физической средой для передачи данных. Это так, особенно в случае, если высокая пропускная способность и низкая стоимость передачи в расчете на один бит являются ключевыми факторами.

Витая пара— два медных изолированных провода, один из которых обвит вокруг другого. Вьющийся провод предназначен для устранения взаимного влияния между соседними витыми парами.

Витая пара широко используется в телефонии. Протяженность линии из витой пары может составлять несколько километров без промежуточного усиления. В России в городских условиях средняя длина абонентской линии около 3,5 км.

Витая пара может быть использована для передачи как цифровых, так и аналоговых сигналов. Ее пропускная способность зависит от толщины используемых проводов и длины линии. На коротких расстояниях (до сотни метров) может достигаться скорость до 1 Тбит/с, на больших расстояниях (несколько километров) скорость передачи не превышает 4 Мбит/с.

Витые пары объединяются в многопарные кабели. Кабель категории 3 содержит четыре витые пары с невысокой плотностью навивки и имеет ширину полосы пропускания до 16 МГц. Наиболее часто используются кабели категории 5, который тоже состоит из четырех пар, но имеет более плотную навивку, что позволяет достигать более высоких скоростей передачи и ширину полосы пропускания 100 МГц.

Частота, МГц	Затухание на каждые 100м, дБ		
	Пара категории 3	Пара категории 5	Экранированная пара (R = 150 Ом)
1	2.6	2.0	1.1
4	5.6	4.1	2.2
16	13.1	8.2	4.4
25	-	10.4	6.2
100	-	22.0	12.3
300	-	-	21.4

Коаксиальные кабели.

У коаксиального кабеля есть два проводника. Центральный проводник представляет собой толстый медный провод, окруженный изолятором. Эта конструкция помещается внутри второго цилиндрического проводника, который обычно представляет собой плетеную плотную металлическую сетку. Все это закрывается плотным защитным слоем пластика. Толщина коаксиального кабеля составляет от 1 до 2,5 см. У такого кабеля шире полоса пропускания и характеристики по затуханию сигнала лучше, чем у витой пары. Коаксиальные кабели работают на частотах от 1 до 500 МГц, поэтому их применяют на больших расстояниях и по ним могут передаваться одновременно несколько потоков данных от разных компьютеров.

Коаксиальные кабели используют для передачи как аналоговых, так и цифровых сигналов. Основными ограничителями скорости и расстояния при передаче без усиления в этих кабелях являются затухание сигнала, тепловой и интермодуляционный шумы. Когда всю полосу пропускания кабеля разбивают на более

узкие полосы и каждую такую полосу используют как отдельный канал, на границах таких каналов возникает интермодуляционный шум.

Оптоволокно.

Для создания оптической связи требуется источник света с постоянной длиной волны, светопроводящая среда и детектор, преобразующий световой поток в электрический. На одном конце оптоволоконной линии имеется передатчик — источник света, световой импульс от которого проходит по светопроводящему волокну и попадает на детектор, расположенный на другом конце этой линии и преобразующий этот импульс в электрический.

Одна из основных проблем при создании оптоволоконных систем состояла в том, чтобы не дать световому пучку рассеяться через боковую поверхность силиконового шнура. Количество рассеиваемой энергии зависит от утла падения светового луча на стенки шнура. При углах больше некоторого критического угла, называемого углом полного внутреннего отражения, вся энергия луча отражается обратно внутрь.

Силиконовый шнур, имеющий толщину, близкую к длине волны источника света, работает без потерь на внутреннее отражение. По такому **одномодовому** шнуру можно передавать данные со скоростью несколько гигабит в секунду на сотню километров без промежуточного усиления.

Поскольку можно испускать несколько лучей разной длины волны так, чтобы они попадали на границы шнура под углом больше, чем угол полного внутреннего отражения, следовательно, по одному шнуру можно пускать несколько лучей. При этом каждый луч, как говорят, имеет свою моду. Так получается **многомодовый** шнур.

Оптоволокно изготавливают из стеклоподобного материала. Затухание оптического сигнала в стекле зависит от длины волны источника света. На практике для передачи сигнала используются три полосы для волн длиной 0,85, 1,30 и 1,55 мкм. Волны длиной 1,30 и 1,55 мкм обладают тем замечательным свойством, что их затухание составляет менее 5 % на километр. Волны длиной 0,85 мкм имеют большее затухание, но они лучше соответствуют возможностям лазерных источников света. Ширина полосы пропускания во всех трех случаях составляет от 25000 до 30000 ГГц.

Другой проблемой при использовании оптоволокна является дисперсия — потеря по мере распространения исходными световыми импульсами начальных форм и размеров. При этом искажения также зависят от длины волны. Одно из возможных решений этой проблемы — увеличение расстояния между соседними сигналами. Однако это сокращает скорость передачи. К счастью, исследования показали, что при придании сигналу некоторой специальной формы дисперсионные эффекты почти исчезают и сигнал можно передавать на тысячи километров. Сигналы, имеющие такую специальную форму, называются **силитонами.**

Оптоволоконный кабель имеет сердечник, состоящий из сверхпрозрачного оптоволокна и изоляционного покрытия. В одномодовом кабеле толщина сердечника составляет 8... 10 мкм, а в многомодовом — 50... 100 мкм. Сердечник имеет оптическое покрытие из стекловолокна с низким коэффициентом рефракции, сокращающего потери света через его границы, и защитное покрытие из пластика. Соединяют его с помощью специальных коннекторов, механически прижимая один край к другому, либо сваркой. При этом в точке соединения теряется от 5 до 20 % мощности сигнала.

Для передачи используются два вида источников света: светодиод (LED) и полупроводниковый лазер, которые обладают разными свойствами. С помощью специальных интерферометров эти источники света можно настроить на требуемую длину волны. На принимающем конце устанавливается фотодиод, время срабатывания которого 1 не, что ограничивает максимальную скорость передачи значением 1 Гбит/с.

Активное подключение содержит промежуточный усилитель электрического сигнала. Фотодиод преобразует оптический сигнал в электрический, который усиливается и передается компьютеру либо транслируется дальше с помощью лазера или светодиода.

1. Ширина полосы пропускания у оптоволокна несравнимо больше, чем у медного кабеля, что позволяет достигать скоростей передачи в сотни гигабит в секунду на расстояниях в десятки километров
2. Оптоволокно компактнее и имеет меньшую массу.
3. Затухание сигнала в оптоволокне существенно меньше, чем в коаксиальном, кабеле и витой паре, и остается постоянным для широкого диапазона частот.
4. Оптоволокно невосприимчиво к внешним электромагнитным излучениям. Следовательно, ему не страшны интерференция, импульсные шумы и взаимные наводки. Оптоволокно не излучает энергию, поэтому не влияет на работу другого оборудования. Его трудно обнаружить, а следовательно, трудно найти и повредить.
5. Чем меньше используется репитеров, тем дешевле система передачи и меньше источников ошибок. С этой позиции оптоволоконные системы достигли большего совершенства. Среднее расстояние между репитерами у них в разы больше, чем у коаксиального кабеля и витой пары.

Носитель	Диапазон	Стандартное	Стандартная	Рестояние между

информации	частот	затухание, дБ/км	задержка, мс/км	репитерами, км
Витая пара	0...3.5 кГц	0.2 (при 1 кГц)	50	2
Многопарный кабель	0...1 МГц	3 (при 1 кГц)	5	2
Коаксиальный кабель	0...500 МГц	7	5	1...9
Оптический кабель	180...370 ТГц	0.2...0.5	5	40

Беспроводная связь востребована для мобильных вычислительных средств и там, где прокладка любого кабеля затруднительна либо невозможна (горы, старые здания), либо если требуется быстрое создание коммуникации.

Электромагнитный спектр.

В вакууме электромагнитная волна распространяется со скоростью света (c = $3*10^8$ м/с). В медном проводнике эта скорость составляет 2/3 от скорости в вакууме. Обозначим f — частоту, λ — длину волны. Фундаментальное соотношение между f, c и λ имеет вид $\boldsymbol{f\lambda = c}$.

При определенных условиях волны распространяются в строго определенном направлении. В этом случае антенна приемника должна быть должным образом ориентирована в пространстве по отношению к антенне передатчика, чтобы принимать сигналы. При других условиях антенна передатчика распространяет электромагнитные волны во всех направлениях.

Для передачи информации из всего спектра частот используют только следующие диапазоны: радиодиапазон, микроволновый, инфракрасный, видимый и частично ультрафиолетовый. Диапазоны рентгеновского излучения, гамма-излучения и большая часть ультрафиолетового, включающие в себя большие частоты, а следовательно, предпочтительные для передачи, требуют, однако, использования сложной аппаратуры для генерации и модуляции, сигналы в них плохо преодолевают препятствия и, что самое главное, они опасны для живой материи.

Количество данных, передаваемых электромагнитной волной, определяется ее шириной, т.е. спектром частот гармоник, составляющих эту волну. При определенных условиях на низких частотах можно закодировать несколько бит данных на 1 Гц, но на высоких частотах это число можно довести до 40 бит. Следовательно, по кабелю с полосой пропускания 500 МГц можно передавать данные со скоростью несколько гигабит в секунду. Учитывая широкую полосу пропускания оптоволоконного кабеля, становится ясно, почему оптоволокно столь привлекательно для сетей ЭВМ.

Задав некоторую полосу длин волн, получим полосу частот, откуда затем найдем скорость передачи для этой полосы частот. Чем шире полоса частот, тем выше битовая скорость, что следует из формулы, связывающей ширину полосы пропускания и битовую скорость передачи.

На практике чаще всего используются узкочастотные полосы передачи.

При широкочастотной передаче, используемой в основном военными и спецслужбами, частота несущей волны изменяется по определенному закону в диапазоне полосы. Перехватить такую передачу можно только в случае, если известен закон изменения частоты несущей.

Радиопередача.

Радиоволны распространяются на большие расстояния, легко преодолевая преграды. Поскольку радиоволны распространяются во всех направлениях, то принимающая и передающая антенны не требуют дополнительной настройки и регулирования взаимного расположения.

Свойства радиоволн зависят от их частоты. На низких частотах, т.е. длинных волнах, они прекрасно преодолевают препятствия, но мощность сигнала падает пропорционально $1/r^3$, где r — расстояние цо источника. На высоких частотах радиоволны распространяются но прямой, но хуже преодолевают препятствия.

На любых частотах радиоволны чувствительны к помехам от электрических устройств. В силу перечисленных причин лицензирование, т.е. право использования частот в радиодиапазоне, находится под жестким контролем государства.

Длинные и средние волны могут огибать поверхность Земли и распространяться на большие расстояния. Короткие полны хотя и поглощаются земной поверхностью, но за счет отражения от ионосферы также могут распространяться на большие расстояния.

Микроволновая передача.

Частоты свыше 10 МГц представляют собой область микроволнового диапазона. Волны в этом диапазоне распространяются в строго определенном направлении и могут быть сфокусированы с помощью параболической антенны, имеющей вид телевизионной тарелки. Однако приемная и передающая антенны при этом должны быть тщательно ориентированы в пространстве по отношению друг к другу. Такая направленность позволяет, построив цепочку ретрансляторов, передавать сигнал на большие расстояния.
Микроволны не проходят сквозь здания так же хорошо, как низкочастотные волны. Кроме того, из-за рефракции в нижних слоях атмосферы они могут отклоняться от прямого направления. При этом увеличивается задержка сигнала и нарушается передача. Помимо того передача на этих частотах зависит и от погоды: при повышении влажности (дождь, туман и т.п.) ширина полосы пропускания резко сужается, растет шум, сигнал рассеивается.
Для увеличения пропускной способности увеличивают частоту, но начиная с частоты 8 ГГц, волны поглощаются водой и, в частности, дождем. Единственный выход из положения в этом случае — изменить маршрут передачи, и обойти область дождя.
Одно из главных достоинств микроволнового диапазона — не требуется прокладка никакой линии. Достаточно получить права на небольшие площадки земли (в сотню квадратных метров) для установки башен-ретрансляторов через каждые 50 км.
Несколько частотных полос в диапазоне 2 400... 2 484 ГГц, например инфракрасные волны, можно использовать свободно без специального разрешения. Однако в разных странах могут использоваться и другие дополнительные диапазоны, например в США помимо указанного диапазона используются диапазоны 902... 928 МГц и 5 725... 5 850 ГГц.

Инфракрасные и миллиметровые волны.
Инфракрасное излучение и излучение в миллиметровом диапазоне используются на небольших расстояниях в блоках дистанционного управления. Основной недостаток таких излучений — они не проходят через преграды. Например, для инфракрасного излучения лист бумаги — непреодолимое препятствие.
Однако этот недостаток одновременно является и преимуществом, поскольку такое излучение в одной комнате не интерферирует с подобным излучением в другой комнате. На использование этих частот не надо также получать разрешение, т. е. это прекрасный канал для передачи данных внутри помещений на небольших расстояниях.

Видимое излучение.
Видимый диапазон также используется для передачи сигналов. Обычно источником света в этом случае является лазер. Монохромное когерентное излучение легко фокусируется. Однако смог, загрязнение атмосферы, дождь или туман портят дело. Передачу такого излучения способны нарушить даже конвекционные потоки воздуха на крыше, возникающие в жаркий день, которые вызывают дрожание луча вокруг приемника, ухудшая качество передачи.

Билет № 19.
TDMA, FDMA, CDMA - методы множественного доступа к беспроводному каналу.

TDMA, FDMA.
Техника разеделения канала на несколько подканалов для предоставления каждому абоненту отдельного подканала называется мультиплексированием, или уплотнением канала. Имеется два основных подхода к мультиплексированию: использование частотного (FDM) и временного (TDM) разделения канала.
При частотном разделении весь диапазон частот полосы пропускания канала разбивается на поддиапазоны - подканалы. По каждому подканалу выполняется передача независимо от того, что происходит в других каналах. При временном разделении используется вся полоса пропускания канала для каждого абонента, но при этом время передачи делится на слоты по числу потенциальных абонентов, и каждому из них выделяется свой интервал времени (слот) для передачи. Частотное разделение хорошо работает в условиях, когда число абонентов фиксированное и каждый из них обеспечивает плотную загрузку канала. При этом каждому абоненту выделяется своя полоса частот, которую он использует независимо от других.
Однако, когда число пользователей велико, их число изменяется или трафик отдельных абонентов нерегулярный, в FDM появляются проблемы. Статическое разделение канала на подканалы является неэффективным решением проблемы доступа при предположении о постоянстве числа абонентов в среднем и нерегулярном трафике у абонентов.
Если каждому пользователю выделить свой слот и тот его не использует, то это будет пустой тратой пропускной способности канала. Таким образом, ни один из известных статических методов не позволяет эффективно распределять нагрузку.

CDMA.
Для многих систем беспроводной связи характерно использование методов множественного доступа FDM, TDM, ALOHA и их комбинаций. Однако такие системы имеют существенные недостатки: ни один из пользователей этих систем не может использовать всю полосу пропускания, предоставленную системе. Если при этом принять в расчет сужение полосы пропускания из-за проблем на границе сот, падение мощности сигналов от мобильных терминалов в пограничных сотовых зонах, накладные расходы на шифрование в целях безопасности, то становится ясно, что высокую скорость передачи в такой системе получить непросто.
Метод множественного доступа на основе разделения кодов — CDMA (Code Division Multiple Access) основан на принципиально иной идее: каждый участник связи может использовать всю полосу пропускания канала в соте за счет применения метода прямого расширения спектра передачи подобно WiFi.
В CDMA-системе каждый бит сообщения кодируется последовательностью из m частиц (чипов). Бит со значением 0 передается инвертированной последовательностью частиц, а бит со значением 1 — прямой. Каждой мобильной станции присваивается уникальный код — последовательность частиц для 0 и для 1. Другими словами, у каждого участника этой системы свой уникальный «язык», поэтому все могут говорить сразу. Понимать друг друга будут только те, кто говорит на одном языке.
Ясно, что такая техника возможна, только если при увеличении объема передаваемой информации будет пропорционально увеличиваться ширина полосы пропускания. При использовании техники FDM канал, равный 1 МГц, можно разделить на 100 подканалов по 10 кГц каждый, и осуществлять передачу по этим подканалам со скоростью 10 Кбит/с (1 бит на 1 Гц). В случае применения CDMA каждый может использовать всю полосу пропускания — 1 МГц, т.е. если использовать 10-разрядные последовательности частиц (что предполагает 2^{10} разных последовательностей), можно передавать данные со скоростью 100 Кбит/с.
Кроме того, поскольку каждая станция имеет уникальную последовательность частиц, не требуется дополнительного шифрования. Из сказанного ясно преимущество CDMA-системы по сравнению с TDM- и FDM-техниками. Идея уникальности последовательности частиц для каждой станции основана на ортогональных кодах. (прим. не автора: в учебнике не особо понятно, но по тексту далее две последовательности ортогональны, если они различаются и совпадают в одинаковом числе разрядов).
Реализация этого элегантного метода потребовала решения целого ряда сложных технических проблем: синхронизация передачи кодовых последовательностей частиц разными станциями; регулирования уровня мощности сигналов в полосе каждой частицы; определения способа, как получатель узнает последовательность частиц отправителя.

Билет № 20.

Телефонные сети: структура, проблема локальной петли (последняя миля). Технологии xDSL.

Структура телефонной сети.

Структура современной телефонной сети весьма избыточная и многоуровневая. Общегосударственная телефонная сеть (ОАКТС) Российской Федерации состоит из междугородной телефонной сети и зоновых телефонных сетей. Междугородная телефонная сеть обеспечивает соединение автоматических междугородных телефонных станций (АМТС) различных зон.

Зоновая телефонная сеть состоит из местных телефонных сетей, расположенных на территории зоны, и внутризоновой телефонной сети, соединяющей между собой эти сети. Местные телефонные сети подразделяются на городские телефонные сети, т. е. обслуживающие город и ближайшие пригороды (ГТС), и сельские, обеспечивающие связь в пределах сельского административного района (СТС).

Учрежденческо-производственная телефонная сеть (УПТС) служит для внутренней связи предприятий, учреждений, организаций и может либо соединяться с сетью общего пользования, либо быть автономной.

Зоновая телефонная сеть включает в себя всех абонентов определенной территории, охватываемой единой семизначной нумерацией, и является частью ОАКТС. Территории зоновых сетей совпадают с территориями административных областей (республик). В зависимости от конфигурации области и телефонной плотности территории нескольких областей могут быть объединены в одну зону, и наоборот, одна область может быть разделена на две зоны и более. Зоновая сеть включает в себя ГТС и СТС, причем на территории одной зоны может быть несколько ГТС и СТС. Крупные города с семизначной нумерацией абонентов, такие как Москва и С.-Петербург, выделяются в отдельные зоны.

Сельские телефонные сети охватывают более обширные территории, чем городские, так как плотность телефонных аппаратов в них значительно меньше. Следовательно, емкость автоматических телефонных станций (АТС) в сельских местностях значительно меньше, чем в городах.

Территория города делится на районы, обслуживаемые районными АТС емкостью от 10000 до 100000 номеров. Протяженность абонентских линий районированной ГТС сокращается, так как АТС приближаются к местам установки телефонных аппаратов. Районные АТС соединяются соединительными линиями (СЛ) по принципу «каждая с каждой».

Рассмотрим структуру телефонного номера, приведенную на рис. 5.3, включающую в себя четыре компонента: код страны, код зоны в стране, затем код территории в зоне (сельского района или крупного города) и только потом номер абонента.

Код страны состоит из кода региона и собственно страны. Регионам присвоены следующие коды:

Северная и Центральная Америка — 1;
Африка — 2
Европа — 3 и 4;
Южная Америка — 5;
страны бывшего СССР — 7;
Центральная Азия и Дальний Восток — 8;
Индия и Ближний Восток — 9.

В каждом из указанных регионов странам присваиваются одно-, двух- и трехзначные коды, первой цифрой в которых является код региона. Например, код 49 соответствует Германии, где 4 — код региона, а 9 — код собственно страны. При этом общее число знаков в телефонном номере не должно превышать 11.

Каждый абонент соединен двумя витыми парами с ближайшей местной телефонной станцией (ТС). Это соединение называется локальным соединением, абонентской линией или последней милей.

Местная ТС соединена в крупных городах с районной ТС либс городской ТС. Районные и городские ТС соединены с региональными или междугородными ТС и т. д.

Если один абонент звонит другому абоненту, который подключен к той же местной ТС, что и звоняший, то коммутаторы этой ТС соединяют абонентов напрямую. Каждая местная ТС соединена с ТС следующего уровня: районными или городскими ТС и междугородними ТС. Если один абонент звонит другому абоненту, телефон которого подключен к другой местной ТС, то местная ТС звонящею соединяется с надлежащей ТС вышерасположенного уровня, которая устанавливает соединение с местной ТС того, кому звонят. В результате создается прямое соединение между абонентами. ТС соединяются между собой магистральными линиями.

Главное следует уяснить, что имеется несколько уровней ТС, каждая из которых может осуществлять коммутацию. Далее телефонные станции любого уровня будем называть просто узлами коммутации. Соединения между узлами коммутации должны обладать большой пропускном способностью, чтобы по ним можно было передавать одновременно несколько разговоров. Пропускная способность местной линии

должна быть достаточной для одного телефонного разговора. Для абонентских линий чаще всего применяли и применяют витую пару. для магистралей между узлами коммутации используют коаксиальные кабели, оптоволокно и радиорелейные линии на микроволнах.
В прошлом телефонная система на всех уровнях была аналоговой, т.е. по проводам передавали колебания напряжения, соответствующие акустическим колебаниям, принимаемым мембраной микрофона. С появлением цифровых методов передачи аналоговая техника стала вытесняться, и в настоящее время аналоговыми остались только абонентские линии, да и то не везде.
Итак, современная телефонная сеть включает в себя:
• абонентскую линию (соединение типа клиент—местная ТС);
• магистрали оптоволоконные или микроволновые (соединение типа ТС—ТС);
• станции коммутации (ТС).

Абонентская линия, или локальное соединение, связывает абонента с ближайшим узлом коммутации. Это соединение также называется последней милей. При передаче данные приходится преобразовывать четыре раза из цифровой формы в аналоговую и обратно. Несмотря на то, что между узлами коммутации передача осуществляется в цифровой форме, в локальном соединении она часто аналоговая.

Последняя миля.
Пропускной способности 3 кГц обычной телефонной абонентской линии со временем стало недостаточно. Возникла проблема, как обеспечить частные квартиры и дома линиями связи надлежащей пропускной способности, — так называемая проблема последней мили.
Работы по решению этой проблемы велись в четырех направлениях. Первое направление называется Fiber To The Home (FTTH) – проведение оптоволокна. Большая пропускная способность, и большая стоимость - это решение имело смысл только для крупных фирм, а не для индивидуальных абонентов.
Второе направление было связано со стремлением сократить длину локального соединения до минимума. Поэтому было предложено протягивать оптоволокно от местного узла коммутации до опорного шкафа развязки внутри микрорайона, а далее были возможны два варианта: использовать от опорного шкафа либо обычную витую пару с технологией HDSL из семейства xDSL, либо коаксиальные кабели сети кабельного телевидения**.** Это решение получило название Hybrid Fiber Coac (HFC).
Третий вариант решения проблемы последней мили — это использование беспроводных технологий Wireless Local Loop, но отметим, что доступный для них диапазон частот сильно ограничен международными соглашениями.
Четвертый вариант решения рассматриваемой проблемы — это использование стандартов серии xDSL.

xDSL family.
Семейство технологий xDSL (Digital Subscriber Line) предназначено для цифровой передачи данных по медной витой паре существующих локальных соединений телефонных кабельных систем. На современном этапе развития семейство xDSL включает в себя следующие технологии: DSL, IDSL,HDSL, SDSL, VDSL, ADSL, RADSL, UADSL.
По аналогии с модемами для работы на физической линии модемы xDSL не ограничиваются передачей информации в спектре телефонных частот, а используют всю полосу пропускания витой пары.
Широкая полоса сигнала, используемого в этом семействе технологий, не позволяет применять такой сигнал в коммутируемых телефонных линиях (телефонные коммутаторы не рассчитаны на такой спектр частот). Модемы xDSL могут работать только на участке телефонных кабельных систем между абонентом и ближайшей телефонной станцией поставщика услуг или между двумя абонентами при непосредственном соединении их абонентских линий (без участия станции коммутации). Это так называемые выделенные линии.
Технологии семейства xDSL используют спектр частот, не пересекающихся со спектром частот телефонного канала, благодаря чему по абонентской линии можно вести телефонные переговоры одновременно с передачей цифровой информации.
Рассмотрим, что представляет собой физический сигнал в xDSL-системах, где используется цифровое кодирование. Стандартный метод цифрового кодирования 2B1Q применяется практически во всех типах оборудования xDSL за исключением технологий ADSL и VDSL.
Гораздо чаще в компьютерных сетях используется метод QAM и его модификация — DMT. Поскольку QAM (Quadrate Amplitude Manipulation) означает квадратурную амплитудную манипуляцию (здесь манипуляция тождественна модуляции, применимой к цифровым сигналам), QAM-сигнал представляет собой сумму двух гармонических колебаний с амплитудами, дискретно изменяющимися и сдвинутыми на 180 градусов относительно друг друга. При этом одну составляющую называют синфазной, а вторую — квадратурной.

Квадратурная амплитудная модуляция обычно является многопозиционной, т. е. за счет множества разрешенных уровней амплитуды за один такт работы модулятора может быть передано несколько бит информации. Число комбинаций (их также называют сигнальными точками) указывается через дефис в обозначении QAM, например: QAM-4, QAM-16, QAM-256. За один цикл работы модулятора QAM-16 передается 4 бит информации, а модулятора QAM-256 — 8 бит.

В настоящее время в подавляющем большинстве xDSL-линий используется DMT-модуляция (Discrete Multi-Tone), при которой все частотное пространство разбивается на 256 каналов шириной по 4 312,5 Гц, и в каждом из этих каналов используется квадратурная амплитудная модуляция с различным числом сигнальных точек.

Для каждого конкретного соединения число сигнальных точек различно, т.е. передающее и приемное устройства сами выбирают число сигнальных позиций в зависимости от наличия в кабеле помех на той или иной частоте. Согласно спецификации стандарта нижние семь каналов вообще никогда не используются, два канала зарезервированы для служебных целей, 25 каналов отводится на восходящий поток и 224 канала, включая служебные, — на нисходящий поток.

Технология	ADSL	HDSL	SDSL	VDSL
Скорость	1.5 – 9 Мбит/с – входящая, 16-640 Кбит/с – исходящая	1.544 или 2.048 Мбит/с	1.544 или 2.048 Мбит/с	13-52 Мбит/с – входящая, 1.5-2.2 Мбит/с - исходящая
Режим	Асимметричный	Симметричный	Симметричный	Асимметричный
Число пар проводов	1	2	1	1
Радиус действия	3.7-5.5 км	3.7 км	3.0 км	1.4 км
Тип сигнала	Аналоговый	Цифровой	Цифровой	Аналоговый
Метод кодирования	DMT	2B1Q	2B1Q	DMT
Частота	1-5 МГц	196 кГц	196 кГц	10 МГц

Билет № 21.

Телефонные сети: структура, локальная петля, магистраль и мультиплексирование.

Структура телефонной сети.

Структура современной телефонной сети весьма избыточная и многоуровневая. Общегосударственная телефонная сеть (ОАКТС) Российской Федерации состоит из междугородной телефонной сети и зоновых телефонных сетей. Междугородная телефонная сеть обеспечивает соединение автоматических междугородных телефонных станций (АМТС) различных зон.

Зоновая телефонная сеть состоит из местных телефонных сетей, расположенных на территории зоны, и внутризоновой телефонной сети, соединяющей между собой эти сети. Местные телефонные сети подразделяются на городские телефонные сети, т. е. обслуживающие город и ближайшие пригороды (ГТС), и сельские, обеспечивающие связь в пределах сельского административного района (СТС).

Учрежденческо-производственная телефонная сеть (УПТС) служит для внутренней связи предприятий, учреждений, организаций и может либо соединяться с сетью общего пользования, либо быть автономной.

Зоновая телефонная сеть включает в себя всех абонентов определенной территории, охватываемой единой семизначной нумерацией, и является частью ОАКТС. Территории зоновых сетей совпадают с территориями административных областей (республик). В зависимости от конфигурации области и телефонной плотности территории нескольких областей могут быть объединены в одну зону, и наоборот, одна область может быть разделена на две зоны и более. Зоновая сеть включает в себя ГТС и СТС, причем на территории одной зоны может быть несколько ГТС и СТС. Крупные города с семизначной нумерацией абонентов, такие как Москва и С.-Петербург, выделяются в отдельные зоны.

Сельские телефонные сети охватывают более обширные территории, чем городские, так как плотность телефонных аппаратов в них значительно меньше. Следовательно, емкость автоматических телефонных станций (АТС) в сельских местностях значительно меньше, чем в городах.

Территория города делится на районы, обслуживаемые районными АТС емкостью от 10000 до 100000 номеров. Протяженность абонентских линий районированной ГТС сокращается, так как АТС приближаются к местам установки телефонных аппаратов. Районные АТС соединяются соединительными линиями (СЛ) по принципу «каждая с каждой».

Рассмотрим структуру телефонного номера, приведенную на рис. 5.3, включающую в себя четыре компонента: код страны, код зоны в стране, затем код территории в зоне (сельского района или крупного города) и только потом номер абонента.

Код страны состоит из кода региона и собственно страны. Регионам присвоены следующие коды:

Северная и Центральная Америка — 1;
Африка — 2
Европа — 3 и 4;
Южная Америка — 5;
страны бывшего СССР — 7;
Центральная Азия и Дальний Восток — 8;
Индия и Ближний Восток — 9.

В каждом из указанных регионов странам присваиваются одно-, двух- и трехзначные коды, первой цифрой в которых является код региона. Например, код 49 соответствует Германии, где 4 — код региона, а 9 — код собственно страны. При этом общее число знаков в телефонном номере не должно превышать 11.

Каждый абонент соединен двумя витыми парами с ближайшей местной телефонной станцией (ТС). Это соединение называется локальным соединением, абонентской линией или последней милей.

Местная ТС соединена в крупных городах с районной ТС либс городской ТС. Районные и городские ТС соединены с региональными или междугородными ТС и т. д.

Если один абонент звонит другому абоненту, который подключен к той же местной ТС, что и звоняший, то коммутаторы этой ТС соединяют абонентов напрямую. Каждая местная ТС соединена с ТС следующего уровня: районными или городскими ТС и междугородними ТС. Если один абонент звонит другому абоненту, телефон которого подключен к другой местной ТС, то местная ТС звонящею соединяется с надлежащей ТС вышерасположенного уровня, которая устанавливает соединение с местной ТС того, кому звонят. В результате создается прямое соединение между абонентами. ТС соединяются между собой магистральными линиями.

Главное следует уяснить, что имеется несколько уровней ТС, каждая из которых может осуществлять коммутацию. Далее телефонные станции любого уровня будем называть просто узлами коммутации. Соединения между узлами коммутации должны обладать большой пропускном способностью, чтобы по ним можно было передавать одновременно несколько разговоров. Пропускная способность местной линии

должна быть достаточной для одного телефонного разговора. Для абонентских линий чаще всего применяли и применяют витую пару. для магистралей между узлами коммутации используют коаксиальные кабели, оптоволокно и радиорелейные линии на микроволнах.
В прошлом телефонная система на всех уровнях была аналоговой, т.е. по проводам передавали колебания напряжения, соответствующие акустическим колебаниям, принимаемым мембраной микрофона. С появлением цифровых методов передачи аналоговая техника стала вытесняться, и в настоящее время аналоговыми остались только абонентские линии, да и то не везде.
Итак, современная телефонная сеть включает в себя:
• абонентскую линию (соединение типа клиент—местная ТС);
• магистрали оптоволоконные или микроволновые (соединение типа ТС—ТС);
• станции коммутации (ТС).

Абонентская линия, или локальное соединение, связывает абонента с ближайшим узлом коммутации. Это соединение также называется последней милей. При передаче данные приходится преобразовывать четыре раза из цифровой формы в аналоговую и обратно. Несмотря на то, что между узлами коммутации передача осуществляется в цифровой форме, в локальном соединении она часто аналоговая.

Последняя миля.
Пропускной способности 3 кГц обычной телефонной абонентской линии со временем стало недостаточно. Возникла проблема, как обеспечить частные квартиры и дома линиями связи надлежащей пропускной способности, — так называемая проблема последней мили.
Работы по решению этой проблемы велись в четырех направлениях. Первое направление называется Fiber To The Home (FTTH) – проведение оптоволокна. Большая пропускная способность, и большая стоимость - это решение имело смысл только для крупных фирм, а не для индивидуальных абонентов.
Второе направление было связано со стремлением сократить длину локального соединения до минимума. Поэтому было предложено протягивать оптоволокно от местного узла коммутации до опорного шкафа развязки внутри микрорайона, а далее были возможны два варианта: использовать от опорного шкафа либо обычную витую пару с технологией HDSL из семейства xDSL, либо коаксиальные кабели сети кабельного телевидения**.** Это решение получило название Hybrid Fiber Coac (HFC).
Третий вариант решения проблемы последней мили — это использование беспроводных технологий Wireless Local Loop, но отметим, что доступный для них диапазон частот сильно ограничен международными соглашениями.
Четвертый вариант решения рассматриваемой проблемы — это использование стандартов серии xDSL.

Магистрали и мультиплексирование.
Созданные в телефонии схемы мультиплексирования можно подразделить на два больших класса: мультиплексирование с разделением частот (Frequency Division Multiplexing) и мультиплексирование с разделением по времени (Time Division Multiplexing). Кроме того, были разработаны методы мультиплексирования на основе разделения длин волн (WDM — Wavelength Division Multiplexing) и на основе разделения кодов (CDM — Code Division Multiplexing). Метод разделения длин волн применяется в оптоволоконных системах. Методы разделения кодов используются в системах беспроводной связи.
Идея мультиплексирования с разделением частот заключается в: весь диапазон частот полосы пропускания кабеля разбивается на поддиапазоны, которые называются каналами. По каждому каналу производится независимая передача.
Объединение 12 голосовых каналов с пропускной способностью по 4000 Гц в полосе от 60 до 108 кГц называется группой. Пять групп по 12 каналов мультиплексируют в супергруппу, а пять супергрупп — в мастергруттпу. Современные стандарты МСЭ позволяют объединять до 230000 голосовых каналов.
Мультиплексирование с разделением волн, используемый для волоконно-оптических каналов, заключается в следующем. Два волоконно-оптических кабеля с импульсами разной длины волны подводятся к одной призме. Свет, пройдя через эту призму (или дифракционную решетку), смешивается в единый луч, который на другом конце разделяется с помощью другой такой же призмы. Поскольку каждый канал занимает лишь несколько гигагерц, а пропускная способность одного оптоволоконного канала около 25000 ГГц (быстрее преобразовывать световой сигнал в электрический пока не получается), то возможности оптоволокна для мультиплексирования огромны. Метод мультиплексирования с разделением длин волн применяется в технологии FTTH.
Мультиплексирование с разделением по времени, или TDM- мультиплексирование (Time Division Multiplexing), предполагает использование цифрового оборудования и хорошо соответствует возможностям компьютера. Следует отметить, что такое мультиплексирование подходит для работы только с данными в

цифровой форме, поэтому, поскольку по абонентской линии телефонный сигнал передается в аналоговой форме, его надо сначала оцифровать.
Оцифровка сигнала происходит на местном узле коммутации, где сходятся абонентские линии с аналоговыми сигналами. На местном узле коммутации аналоговые сигналы с абонентских линий оцифровываются, объединяются и передаются на узлы коммутации следующего уровня по магистральным шинам. Рассмотрим, как это происходит.
Стандарт E1 предполагает мультиплексирование 30 каналов. Каждая из 30 линий сканируется с частотой 8 кГц. Результаты каждого измерения представляют 8-битовое число. Это означает, что в методе ИКМ используются 256 уровней. В стандарте T1 используют 7 бит, т.е. 128 уровней.
Полученные 240 бит упаковывают в кадр. Кадр в стандарте E1 содержит 32 канала по восемь разрядов и занимает 125 мкс. При этом 30 каналов используется для передачи данных, а два — для целей управления. Таким образом, стандарт E1 обеспечивает скорость передачи 2,048 Мбит/с и мультиплексирует 30 линий одновременно.
Стандарт T1, позволяет мультиплексировать 24 линии, но в каждом канале под данные используется лишь семь разрядов и один разряд для целей управления. Кадр в стандарте T1 содержит 193 бит и занимает те же 125 мкс, что и в стандарте Б1, обеспечивая скорость передачи 1,544 Мбит/с. Отметим также, что в E1 из 256 бит кадра 16 бит используются для служебных целей, а в T1 из 193 бит для служебных целей используются 24 бит, т.е. E1 экономичней.
TDM-мультиплексирование позволяет мультиплексировать уже мультиплексированные каналы. Так, согласно стандарту T1 четыре канала уровня T1 можно объединить в один канал уровня T2, затем шесть каналов уровня T2 объединить в один канал уровня T3 и семь каналов уровня T3 — в один канал уровня T4. В результате на уровне T4 максимальная скорость передачи будет равна 274,176 Мбит/с.
Согласно стандарту E1 группировать можно только четыре канала, однако в этом случае будет четыре уровня вложенности, а не три, как в стандарте T1, поэтому скорость передачи составит на уровне E1 2,048, на E2 - 8,848, на E3 - 34,304, на E4 - 139,264, а на E5 - 565,148 Мбит/с.
Создание стандарта SONET (Synchronous Optical NETwork) преследовало четыре основные цели:
• обеспечить возможность использования разных физических сред в телефонной сети, что требовало проработки стандартов кодировки на физическом уровне, выбора длины волны, частоты, временных характеристик сигналов, структуры кадра;
• унифицировать американские, европейские и японские цифровые системы, в которых используются каналы с пропускной способностью 64 Кбит/с с импульсно-кодовой модуляцией;
• обеспечить иерархическое мультиплексирование нескольких цифровых каналов (на сегодня его используют до уровня T3, хотя стандарт определяет и уровень T4);
• определить правила функционирования, администрирования и поддержки оптических каналов связи,
С самого начала было принято решение использовать в стандарте SONET традиционное TDM-мультиплексирование, где вся ширина оптоволоконной линии используется под один канал, содержащий временные слоты подканалов.
Биты на линии SONET имеют строго выверенную длительность, контролируемую главными часами.
Система передачи данных SONET состоит из коммутаторов, мультиплексоров и повторителей, соединенных оптическими линиями. В терминологии SONET сплошной фрагмент оптоволоконного кабеля между двумя устройствами называется секцией. Канал между двумя мультиплексорами (возможно с несколькими повторителями между ними) называется линией. Канал между двумя оконечными абонентами называется путем.
Кадр SONET содержит 810 байт и занимает 125 мкс. Стандарт SONET допускает разные топологии каналов связи, но чаще это двунаправленное кольцо. Так как система SONET синхронная, то кадры генерируются строго один за другим без перерывов вне зависимости от того, имеются данные для передачи или нет. Скорость передачи 8000 кадров/с как раз соответствует каналам с ИКМ-модуляцией, используемым в цифровой телефонии. Исходя из этого, нетрудно подсчитать, что пропускная способность канала SONET составляет 51,84 Мбит/с.
Для описания кадра SONET представим его 810 байт в виде матрицы из 9 строк и 90 столбцов. Каждый элемент такой матрицы — один байт. Первые три элемента в каждой строке — это служебная информация, используемая для администрирования и управления передачей. Первые три элемента первых трех строк образуют заголовок секции, а каждые три первых элемента в следующих шести строках — заголовок линии. Заголовки секции генерируются и проверяются в начале и в конце каждой секции. Аналогично генерируются и проверяются заголовки на каждой линии. Скорость передачи 8000 кадров/с соответствует основному каналу, называемому Synchronous Transport Signal-1 (STS-1).
Оставшиеся в 87 столбцах и 9 строках 783 байт приходятся на данные пользователей, которые образуют так называемый SPE-конверт (Synchronous Payload Envelope), содержащий как данные пользователя, так и

служебную информацию, которая занимает 13 байт SPE-конверт может начинаться с любого байта в оставшихся 9 х 87 байтах. Начало SPE-конверта указано в первых двух байтах третьей строки. Учитывая, что в SONET генерируется 8000 кадров в секунду, получаем, что полезная пропускная способность составит 8000 * 783 * 8 = 50.112 Мбит/с. Нетрудно увидеть, что такая организация работы канала предполагает плотную его загрузку со стороны абонентов.

Мультиплексирование множественных потоков данных, называемых в системе SONET притоками, происходит побайтно. Например, когда три притока STS-1, каждый из которых имеет скорость 51,84 Мбит/с, объединяются в один приток STS-3 со скоростью 155,52 Мбит/с, мультиплексор сначала берет 1-й байт 1-го притока, затем 1-й байт 2-го притока, затем 1-й байт 3-го, и только после этого он переходит к 2-м байтам этих притоков. Кадр STS-3 содержит 270 х9 = 2 430 байт и занимает 125 мкс. Таким образом, на этом уровне битовая скорость составляет 155,52 Мбит/с.

Билет № 22.
Телефонные сети: структура, методы коммутации.

Структура телефонной сети.
Структура современной телефонной сети весьма избыточная и многоуровневая. Общегосударственная телефонная сеть (ОАКТС) Российской Федерации состоит из междугородной телефонной сети и зоновых телефонных сетей. Междугородная телефонная сеть обеспечивает соединение автоматических междугородных телефонных станций (АМТС) различных зон.
Зоновая телефонная сеть состоит из местных телефонных сетей, расположенных на территории зоны, и внутризоновой телефонной сети, соединяющей между собой эти сети. Местные телефонные сети подразделяются на городские телефонные сети, т. е. обслуживающие город и ближайшие пригороды (ГТС), и сельские, обеспечивающие связь в пределах сельского административного района (СТС).
Учрежденческо-производственная телефонная сеть (УПТС) служит для внутренней связи предприятий, учреждений, организаций и может либо соединяться с сетью общего пользования, либо быть автономной.
Зоновая телефонная сеть включает в себя всех абонентов определенной территории, охватываемой единой семизначной нумерацией, и является частью ОАКТС. Территории зоновых сетей совпадают с территориями административных областей (республик). В зависимости от конфигурации области и телефонной плотности территории нескольких областей могут быть объединены в одну зону, и наоборот, одна область может быть разделена на две зоны и более. Зоновая сеть включает в себя ГТС и СТС, причем на территории одной зоны может быть несколько ГТС и СТС. Крупные города с семизначной нумерацией абонентов, такие как Москва и С.-Петербург, выделяются в отдельные зоны.
Сельские телефонные сети охватывают более обширные территории, чем городские, так как плотность телефонных аппаратов в них значительно меньше. Следовательно, емкость автоматических телефонных станций (АТС) в сельских местностях значительно меньше, чем в городах.
Территория города делится на районы, обслуживаемые районными АТС емкостью от 10000 до 100000 номеров. Протяженность абонентских линий районированной ГТС сокращается, так как АТС приближаются к местам установки телефонных аппаратов. Районные АТС соединяются соединительными линиями (СЛ) по принципу «каждая с каждой».
Рассмотрим структуру телефонного номера, приведенную на рис. 5.3, включающую в себя четыре компонента: код страны, код зоны в стране, затем код территории в зоне (сельского района или крупного города) и только потом номер абонента.
Код страны состоит из кода региона и собственно страны. Регионам присвоены следующие коды:
Северная и Центральная Америка — 1;
Африка — 2
Европа — 3 и 4;
Южная Америка — 5;
страны бывшего СССР — 7;
Центральная Азия и Дальний Восток — 8;
Индия и Ближний Восток — 9.

В каждом из указанных регионов странам присваиваются одно-, двух- и трехзначные коды, первой цифрой в которых является код региона. Например, код 49 соответствует Германии, где 4 — код региона, а 9 — код собственно страны. При этом общее число знаков в телефонном номере не должно превышать 11.
Каждый абонент соединен двумя витыми парами с ближайшей местной телефонной станцией (ТС). Это соединение называется локальным соединением, абонентской линией или последней милей.
Местная ТС соединена в крупных городах с районной ТС либс городской ТС. Районные и городские ТС соединены с региональными или междугородными ТС и т. д.
Если один абонент звонит другому абоненту, который подключен к той же местной ТС, что и звоняший, то коммутаторы этой ТС соединяют абонентов напрямую. Каждая местная ТС соединена с ТС следующего уровня: районными или городскими ТС и междугородними ТС. Если один абонент звонит другому абоненту, телефон которого подключен к другой местной ТС, то местная ТС звонящею соединяется с надлежащей ТС вышерасположенного уровня, которая устанавливает соединение с местной ТС того, кому звонят. В результате создается прямое соединение между абонентами. ТС соединяются между собой магистральными линиями.
Главное следует уяснить, что имеется несколько уровней ТС, каждая из которых может осуществлять коммутацию. Далее телефонные станции любого уровня будем называть просто узлами коммутации. Соединения между узлами коммутации должны обладать большой пропускном способностью, чтобы по ним можно было передавать одновременно несколько разговоров. Пропускная способность местной линии

должна быть достаточной для одного телефонного разговора. Для абонентских линий чаще всего применяли и применяют витую пару. для магистралей между узлами коммутации используют коаксиальные кабели, оптоволокно и радиорелейные линии на микроволнах.
В прошлом телефонная система на всех уровнях была аналоговой, т.е. по проводам передавали колебания напряжения, соответствующие акустическим колебаниям, принимаемым мембраной микрофона. С появлением цифровых методов передачи аналоговая техника стала вытесняться, и в настоящее время аналоговыми остались только абонентские линии, да и то не везде.
Итак, современная телефонная сеть включает в себя:
• абонентскую линию (соединение типа клиент—местная ТС);
• магистрали оптоволоконные или микроволновые (соединение типа ТС—ТС);
• станции коммутации (ТС).

Абонентская линия, или локальное соединение, связывает абонента с ближайшим узлом коммутации. Это соединение также называется последней милей. При передаче данные приходится преобразовывать четыре раза из цифровой формы в аналоговую и обратно. Несмотря на то, что между узлами коммутации передача осуществляется в цифровой форме, в локальном соединении она часто аналоговая.

Последняя миля.
Пропускной способности 3 кГц обычной телефонной абонентской линии со временем стало недостаточно. Возникла проблема, как обеспечить частные квартиры и дома линиями связи надлежащей пропускной способности, — так называемая проблема последней мили.
Работы по решению этой проблемы велись в четырех направлениях. Первое направление называется Fiber To The Home (FTTH) – проведение оптоволокна. Большая пропускная способность, и большая стоимость - это решение имело смысл только для крупных фирм, а не для индивидуальных абонентов.
Второе направление было связано со стремлением сократить длину локального соединения до минимума. Поэтому было предложено протягивать оптоволокно от местного узла коммутации до опорного шкафа развязки внутри микрорайона, а далее были возможны два варианта: использовать от опорного шкафа либо обычную витую пару с технологией HDSL из семейства xDSL, либо коаксиальные кабели сети кабельного телевидения**.** Это решение получило название Hybrid Fiber Coac (HFC).
Третий вариант решения проблемы последней мили — это использование беспроводных технологий Wireless Local Loop, но отметим, что доступный для них диапазон частот сильно ограничен международными соглашениями.
Четвертый вариант решения рассматриваемой проблемы — это использование стандартов серии xDSL.

Коммутация.
Третьим важным компонентом телефонной сети являются телефонные станции, или, как их еще называют, узлы коммутации, основу которых составляют коммутаторы. В телефонных сетях используются два разных способа коммутации: коммутация каналов и коммутация пакетов.
Коммутаторы прямые и каскадные.
Самым простым видом является прямой коммутатор типа n x n, т.е. коммутатор, имеющий n входных и n выходных линий, в точках пересечения которых установлены полупроводниковые переключатели, обеспечивающие замыкание соответствующих линий.
Основной недостаток коммутаторов этого типа— квадратичный рост их сложности при увеличении числа линий т.е. сложность коммутатора определяется числом точек пересечения (соединения) этих линий. Даже с учетом того, что для дуплексных линий и при отсутствии самосоединений для работы требуется только половина соединений (выше или ниже главной диагонали), то все равно необходимо иметь порядка n(n-1)/2 переключателей. При n = 1000 на кристалле можно поместить такое количество переключателей, но при этом у него должно быть 2000 ножек, что обеспечить очень не просто. Поэтому такие прямолинейные решения схем коммутаторов возможны лишь для небольших организаций.
Идея построения каскадных коммутаторов заключается в разделении прямого коммутатора на части и соединении этих частей между собой промежуточными дополнительными коммутаторами.
Рассмотрим для примера трехслойный каскадный коммутатор. В первом слое такого коммутатора N входных линий разбивают на группы по n линий в каждой группе. Каждую из N/n групп обслуживает прямой коммутатор n x k**.** Во втором слое k прямых коммутаторов обеспечивают соединение Nk/n входных линий с Nk/n выходными линиями. Третий слой состоит из N/n прямых коммутаторов, каждый из которых соединяет к линий с n линиями. Подсчитаем сложность такого каскадного коммутатора. Число точек пересечения в первом каскаде определяется по формуле (N/n)nk = Nk.

Второй каскад содержит $k(N/n)^2$ точек пересечения, а третий каскад по сложности такой же, как и первый. Таким образом, число точек пересечения в данном коммутаторе составляет $2kN+ k(N/n)^2$**.** Следовательно, при N = 1000, n = 50 и к = 10 для работы потребуется всего 24000 точек пересечения, а не 499 500, как при использовании прямого коммутатора.

Недостатком каскадных коммутаторов является блокировка коммутаторов второго слоя.

В 1953 г. Клос показал, что при k = 2n - 1 блокировок в каскадных коммутаторах не будет.

Коммутаторы с разделением времени.

Пусть имеется n линий, которые необходимо коммутировать. Эти линии сканируют последовательно одна за другой в течение определенного временного слота.

Образуется кадр из n ячеек по k бит в каждой. Например, в стандарте E1 каждая ячейка содержит 8 бит, кадр — 32 ячейки, а всего за секунду проходит 8000 кадров.

Затем кадр попадает в коммутатор ячеек, который переставляет ячейки в соответствии с таблицей коммутации. Обработка кадра происходит следующим образом. Входной кадр записывается в память в том порядке, в каком ячейки считывались с линий. Затем ячейки считываются из памяти в порядке, задаваемом таблицей коммутации.

Ясно, что таблица коммутации — это вектор перестановок, а скорость коммутации ограничена скоростью считывания из памяти. Например, если временной слот составляет 125 мкс, и требуется обработать кадр из n ячеек, а время считывания из памяти T, мкс, то $2nT= 125$ мкс или $n = 125/(2T)$. Если скорость работы памяти 100 нс, то можно обработать не более 625 линий.

Билет 23.
Принципы построения и архитектура СПД ISDN.

ISDN задумывалась как всемирная телекоммуникационная сеть, которая должна была заменить телефонные сети. При этом приложения ISDN должны были поддерживать передачу голоса, звука, изображения и данных. ISDN-телефон по замыслу проекта должен был обеспечивать самый разнообразный сервис: различные программируемые функции, автоматическое определение номера телефона, с которого поступил звонок, и имени звонящего, взаимодействие с компьютером и т. д. Эта технология должна была предусматривать подключение цифровых приборов и оборудования прямо к СПД, т.е. без использования модемов.

Принципы организации СПД ISDN
1. *Поддержка голосовых и не голосовых приложений с использованием определенного набора стандартизованных средств.* Этот принцип определяет цели ISDN и средства их достижения. СПД ISDN поддерживает разнообразные сервисы, т. е. и голосовую связь (телефон), и не голосовую связь (обмен данными в цифровой форме).

2. *Поддержка приложений, использующих как коммутируемые, так и некоммутируемые каналы.* В ISDN используется и коммутация каналов, и коммутация пакетов. СПД ISDN поддерживает приложения, в которых используется:

- выделенные цифровые каналы;
- коммутируемые телефонные сети общего пользования;
- сеть передачи данных с коммутацией каналов;
- сеть передачи данных с коммутацией пакетов;
- сеть передачи данных с трансляцией кадров (FR).

3. *Ориентация на соединения с пропускной способностью 64 Кбит/с.* ISDN-соединения, основанные как на коммутации каналов, так и на коммутации пакетов, должны обеспечивать скорость передачи 64 Кбит/с.

4. *Интеллектуальные сети.* СПД ISDN должна поддерживать сервис высокого уровня: например выполнять переадресацию звонков, автоматически определять разные виды терминалов и т.д.

5. *Использование уровневой архитектуры.* Протоколы доступа к СПД ISDN должны иметь уровневую архитектуру, соответствующую OSI-модели, что обеспечивает целый ряд преимуществ:

- для OSI-модели уже создано много стандартов, например протокол HDLC;
- возможность создания новых ISDN-стандартов на основе уже существующих, а следовательно, сокращение стоимости их реализации;
- возможность независимого развития и реализации стандартов разных уровней.

6. *Использование разнообразных конфигураций каналов и физического оборудования.* Это обеспечивает приспособляемость ISDN к различиям технологической политики в разных странах, уровней технологий и имеющегося оборудования.

Архитектура СПД ISDN

Основой ISDN-архитектуры является концепция *битового потока в цифровом тракте,* или просто *цифрового тракта. Б*иты можно передавать по тракту в обоих направлениях.
Цифровые тракты могут мультиплексировать с разделением по времени в несколько независимых цифровых потоков. Концепция цифрового тракта строго специфицирована. В спецификации определены интерфейсы, формат цифрового потока и правила мультиплексирования потоков. Причем были разработаны два стандарта: один для низкоскоростной передачи, а второй для высокоскоростной
Битовый тракт в ISDN может быть мультиплексирован по нескольким стандартным каналам:

- A — стандартный аналоговый телефонный канал;
- B — цифровой канал с импульсно-кодовой модуляцией для голоса или данных на 64 Кбит/с;
- D — цифровой канал на 16 или 64 Кбит/с;
- H - цифровой канал на 384 (НО), 1536 (НИ), 1 920 (Н12) Кбит/с,

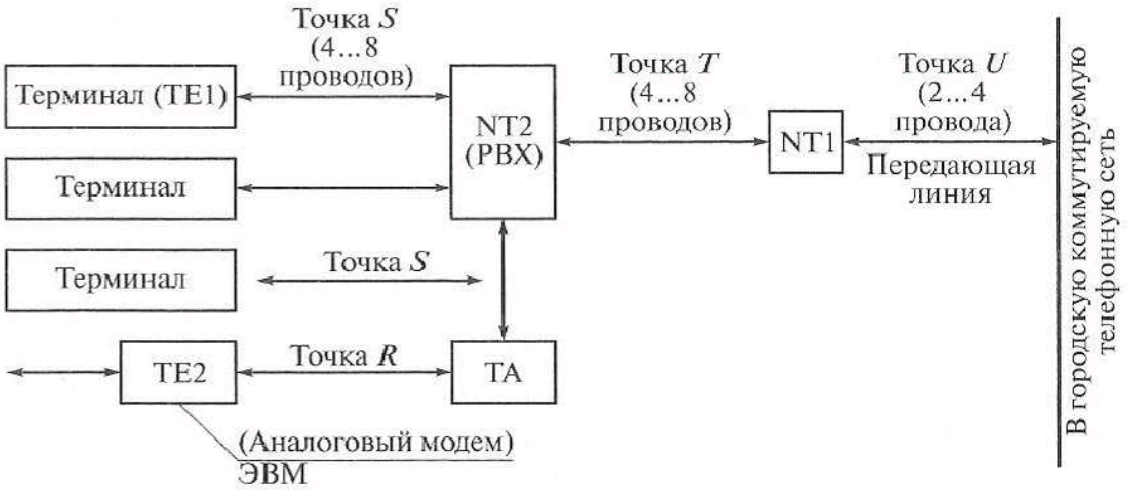
Канал типа B предназначен для поддержки следующих четырех
видов соединений:

- *с коммутацией каналов.* Абонент инициирует вызов, под воздействием которого устанавливается соединение в СПД с коммутацией каналов, например в телефонной сети между двумя абонентами. По созданному каналу передается битовый поток;
- *с коммутацией пакетов.* Абонент подключен к узлу СПД с коммутацией пакетов и обменивается данными с другими абонентами, например посредством протоколов семейства X.25;
- *Frame Relay.* Абонент подсоединяется к узлу СПД Frame Relay, через который происходит обмен данными;
- *постоянное.* Это соединение с другим абонентом, которое было установлено заранее и динамически изменено быть не может, т.е. это соединение, подобное выделенной линии.

Канал типа D, во-первых, служит для управления коммутацией каналов, инициированной вызовом по интерфейсу с абонентом через канал B, а во-вторых, может использоваться, когда свободен, для коммутации пакетов или получения данных от оборудования на низкой скорости (до 100 бит/с). Канал типа H служит для высокоскоростной передачи данных. Абонент может использовать такой канал как высокоскоростную магистраль либо разделить ее с помощью метода TDM на подканалы.
Обычно каналы этого типа используются для таких приложений, как факс, видео, высококачественные звуковые устройства. Для подключения абонентов указанные каналы объединяются в так называемые интерфейсные структуры, или интерфейсы. В настоящее время лучше всего определены и часто используются два интерфейса: базовый и основной. *Базовый интерфейс* (BRI — Basic Rate Interface), или *базовый доступ* (BA), состоит из двух каналов типа B и одного канала типа D.
Основной интерфейс (PRI), или *основной доступ* (PA — Primary Access), предназначен для пользователей с большими требованиями к пропускной способности сети (1 544 Кбит/с и более). Существует два варианта основного интерфейса: для передачи со скоростью 1 544 Кбит/с, что соответствует стандарту T1, и для передачи со скоростью 2 048 Кбит/с, что соответствует стандарту E1. В первом варианте основной интерфейс включает в себя 23 B-канала и один D-канал, а во втором — 30 B-каналов и один D-канал. При этом по всем каналам передача данных идет синхронно. Длина кадра у базового интерфейса равна 48 разрядам: по 16 разрядов на каждый B-канал и 4 бит для D-канала. Остальные 12 разрядов носят служебный характер. Кадр основного интерфейса для передачи со скоростью 1,544 Мбит/с имеет длину 193 бит и занимает 125 мкс, а для передачи со скоростью 2,048 Мбит/с •- длину 256 бит и занимает те же 125 мкс. Базовый интерфейс (BRI) может поддерживать не только схему 2B + D, но и схемы B + D и просто D. Возможны варианты интерфейса PRI с меньшим числом каналов типа B, например 20B + D. Каналы типа B могут объединяться в один логический высокоскоростной канал для передачи с общей скоростью до 1 920 Кбит/с. При установке у пользователя нескольких интерфейсов PRI все они могут иметь один канал типа D. Число каналов B в интерфейсе, который не имеет канала D, может увеличиваться до 24 или 31.
Основной интерфейс может быть создан на основе каналов типа H, При этом общая пропускная способность такого интерфейса все равно не должна превышать 2,048 или 1,544 Мбит/с. Для каналов типа HO возможны следующие схемы интерфейсов: 3H0 + D — американский вариант и 5HO + D — европейский. Для каналов *H1x* возможен интерфейс, состоящий только из одного канала НИ (1,536 Мбит/с) — американский вариант или одного канала H12 (1,920 Мбит/с) и одного канала D — европейский вариант.

Билет № 24.
Подключение оборудования пользователей в СПД ISDN.

Подключение пользовательского оборудования к СПД ISDN осуществляется в соответствии со схемой, в которой все оборудование пользователей подразделяется на функциональные группы. В зависимости от функциональных групп различают несколько типов точек подключения (reference points) оборудования.



Устройства функциональной группы NT1 (Network Termination 1) подключаются через точку типа ***U.*** Фактически NT1 представляет собой устройство, которое работает на физическом уровне и образует дуплексный канал с соответствующим устройством, установленным на территории оператора СПД ISDN. Если пользователь подключен через интерфейс BRI, то абонентская линия — обычное окончание аналоговой телефонной сети, т.е. витая пара, используемая по стандарту DSL BRI. Максимальная длина абонентской линии в этом случае составляет 5,5 км. При использовании интерфейса PRI абонентская линия представляет собой четырехпроводную линию, функционирующую по стандарту T1 или E1, с максимальной длиной около 1 800 м.

Устройства функциональной группы NT2 (Network Termination 2) выполняют функции концентратора битовых потоков от оборудования пользователя и осуществляют их мультиплексирование. К этому типу оборудования относятся: офисная АТС (PBX), обеспечивающая коммутацию каналов; коммутатор пакетов (например, по каналу D); простой мультиплексор TDM, который мультиплексирует несколько низкоскоростных битовых потоков в один канал типа B. Оборудование типа NT2 подключается к устройству NT1 через точку типа T. Наличие оборудования этого типа не является обязательным в СПД ISDN в отличие от оборудования типа NT1.

Устройства функциональной группы TE1 (Terminal Equipment I) поддерживают интерфейс пользователя BRI или PRI. Точка подключения типа S соответствует точке подключения отдельного терминального оборудования, поддерживающего один из интерфейсов пользователя СПД ISDN, например цифрового телефона или факс-аппарата. Так как оборудование типа NT2 может отсутствовать в СПД ISDN, то точки подключения типов ***S*** и ***T*** объединяются и обозначаются как S/T.

Устройства функциональной группы TE2 (Terminal Equipment 2) не поддерживают интерфейсы BRI и PRI. К таким устройствам может относиться компьютер и устройства с последовательными интерфейсами, не относящимися к ISDN, например RS-232C, X.21 или V.35. Для подключения такого устройства к СПД ISDN необходимо использовать терминальный адаптер (Terminal Adaptor — TA). Для компьютеров терминальные адаптеры выпускаются в формате сетевых адаптеров, т.е. в виде встраиваемой карты. Физически интерфейс в точке S/T представляет собой четырехпроводную линию. Кабель между устройством TE1 или TA и сетевым окончанием NT1 или NT2 обычно небольшой длины, поэтому разработчики стандартов ISDN решили не усложнять оборудование, поскольку организация дуплексного режима в четырехпроводной линии намного легче, чем в двухпроводной. Для интерфейса BRI в качестве метода кодирования используется биполярный AMI, в котором логическая единица кодируется нулевым потенциалом, а логический нуль — чередованием потенциалов противоположной полярности. Для интерфейса PRI используются те же коды, что и для интерфейсов T1 и E1.

Билет № 25.
Стек протоколов СПД ISDN.

В СПД ISDN есть два стека протоколов: стек каналов типа D и стек каналов типа B.
Каналы типа D образуют достаточно традиционную СПД с коммутацией пакетов, прообразом которой послужила СПД X.25. Для сети каналов типа D установлены три уровня протоколов: физический протокол, определяемый стандартом I.430/431; канальный протокол LAP-D, определяемый стандартом Q.921, а на пакетном уровне протокол Q.931, с помощью которого выполняется коммутация вызова абонента службы с коммутацией каналов, или протокол X.25, при использовании которого в кадры протокола LAP-D вкладываются пакеты X.25 и коммутаторы ISDN выполняют роль коммутаторов X.25.
Для мониторинга и управления СПД ISDN сеть каналов типа D использует так называемую *систему сигнализации номер* 7 (Signal System Number 7 — SS7). Эта система, разработанная для внутреннего мониторинга и управления коммутаторами телефонной сети общего назначения, здесь рассматриваться не будет. Однако отметим, что конечному пользователю услуги службы SS7 недоступны, так как сообщениями SS7 коммутаторы сети обмениваются только между собой.
Каналы типа B образуют СПД с коммутацией цифровых каналов. В терминах модели OSI на основе каналов типа B в коммутаторах СПД ISDN определен только протокол физического уровня I.430/431. Коммутация каналов типа B происходит по указаниям, полученным по каналу D. Когда пакеты протокола Q.931 маршрутизируются коммутатором, происходит одновременная коммутация очередной части составного канала типа B от исходного абонента к конечному.
Протокол LAP-D, являющийся аналогом протокола LAP-B в СПД X.25, принадлежит семейству HDLC и обладает всеми родовыми чертами этого семейства, но имеет некоторые особенности. Адрес кадра LAP-D состоит из двух байтов, один из которых определяет код службы, куда пересылаются вложенные в кадр пакеты, а второй — используется для адресации одного из терминалов, подключаемых у пользователя к абонентской линии NT1. Терминальное устройство может поддерживать разные службы: службу установления соединения по протоколу Q.931, службу коммутации пакетов X.25, службу мониторинга СПД и т.д. Протокол LAP-D обеспечивает два режима работы: с установлением соединения (единственный режим работы протокола LLC2) и без установления соединения (режим LLC1 стандарта IEEE 802.1). Последний режим используется, например для управления и мониторинга СПД.

Билет № 26.
Передача данных в ATM сетях.

В сетях ATM нет строго порядка поступления ячеек от различных источников. Ячейки могут поступать от разных источников и в разном порядке. Необязательно даже, чтобы поток ячеек от одного источника был непрерывен. Если возникают разрывы, то они заполняются ячейками ожидания. В стандарте ATM требуется, чтобы ячейки могли передаваться носителями (кадрами, фреймами и т.п.) в рамках таких стандартов, как T1, T3, El, SONET, FDDI и некоторых других. В настоящее время скорость 155,52 Мбит/с является стандартной для сетей ATM, так же как и учетверенная скорость 622,08 Мбит/с. Стандартной средой передачи для ATM является оптоволокно. Однако на расстояниях в сотни метров в них может использоваться коаксиальный кабель или витая пара 5-й категории. Применение оптоволокна обеспечивает передачу на расстоянии многих километров. Каждая волоконно-оптическая линия соединяет либо компьютер с ATM-переключателем, либо два ATM-переключателя. ATM- линии — это соединения типа точка—точка. При этом на одной линии не может находиться более одного источника ячеек. По каждой линии передача возможна только в одном направлении, поэтому для обеспечения полного дуплекса необходимы две ATM-линии. С помощью ATM-переключателей возможно дублирование одной и той же ячейки для передачи ее по нескольким линиям. Так реализуется режим вещания.

ATM-переключатель имеет набор входных линий, по которым в него поступают ячейки, и, как правило, такое же число исходящих линий, по которым ячейки проходят после коммутации. Обычно переключатель работает синхронно: длительность цикла строго фиксирована. В течение каждого цикла просматриваются все входные линии и, если на линию к этому моменту целиком поступила ячейка, то она считывается и передается в центр коммутации, а затем на выходную линию. Переключатель может быть конвейерным, т.е. обработка одной ячейки может занимать более одного цикла. Ячейки поступают асинхронно, таймер переключателя отмечает момент начала очередного цикла. Если ячейка не поступила целиком к началу цикла, то она должна ожидать начала следующего цикла. Ячейки поступают со скоростью 155 Мбит/с. Учитывая что размер ячейки 53 байт, получаем около 360 000 ячеек/с. Выпускаемые в настоящее время переключатели имеют от 16 до 1 024 входных линий
Все ATM-переключатели должны удовлетворять следующим требованиям:
• терять как можно меньше ячеек (т.е. обеспечивать достаточно большую скорость переключения без потери ячеек; допустима потеря одной ячейки на каждые 1012 или потеря одной-двух ячеек за час работы);
• никогда не менять порядок поступления ячеек по каждому виртуальному соединению.

Что делать, когда сразу по нескольким линиям пришли ячейки, которые должны быть отправлены по одной и той же выходной линии. Возможное решение: *буферизовать ячейки на входе переключателя*. Необходимо следить за тем, чтобы дисциплина обслуживания возникающих очередей была справедливой, т.е. чтобы равномерно обслуживались очереди на всех линиях. Недостаток: *блокировка на входе*: очередь на входе может блокировать даже те ячейки, которые должны быть перекоммутированы на линии, где нет конфликтов. Кроме того, буферизация ячейки на входе требует дополнительной логики в схемах, что усложняет конструкцию ATM-переключателя. Альтернативное решение: *буферизация ячеек на выходе переключателя* (*переключатели выталкивающего типа*). Если несколько ячеек должны уйти по одной и той же линии, то они передаются на выход переключателя и там их буферизуют. Буферизация на выходе эффективнее, чем буферизация на входе.

Основным недостатком переключателей выталкивающего типа является то, что центр коммутации — это прямой коммутатор, а следовательно, его сложность растет квадратично по отношению к числу коммутируемых линий. Одним из решений этой проблемы является использование каскадных коммутаторов. Результат такого решения называется *баньяновым переключателем Батчера*. Баньяновый переключатель Батчера синхронный, т.е. за один цикл он может обрабатывать несколько входных линий. В баньяновых переключателях для каждого входа существует только один путь к любому из выходов. Маршрутизация пакета происходит в каждом узле на основе адреса выходной линии, которую должен достичь пакет. Адрес выходной линии определяется на входе по номеру виртуального соединения. Если две ячейки, поступившие на вход одного и того же коммутирующего элемента, должны быть направлены в один и тот же порт, то одна из них туда направляется, а другая — сбрасывается. Адрес выходной линии анализируется в каждом элементе слева направо. Например, код 001 означает, что соответствующая ячейка будет направлена сначала в верхний порт, потом еще раз в верхний, а только затем в нижний.

В зависимости от распределения ячеек на входе баньяновая сеть либо будет терять ячейки, либо нет. Идея Батчера состояла в том, чтобы переставить ячейки на входах таким образом, чтобы в баньяновой сети конфликтов не возникало.
Для сортировки входов Батчер в 1968 г. предложил специальный переключатель. Подобно баньяновому переключателю переключатель Батчера строится из элементов по схеме 2 x 2, а работает синхронно и дискретно. В каждом элементе этого переключателя выходные адреса ячеек сравниваются. При этом ячейка с большим адресом направляется по стрелке, а с меньшим — в противоположном направлении. Если ячейка одна, то она направляется против стрелки. Сравниваются не отдельные биты, а целые адреса как число.
Известны две проблемы, которые баньяновые переключатели Батчера не могут решить:
• при возникновении коллизии на их выходе возможен только сброс;
• рассылка одной и той же ячейки сразу на несколько выходов.

Уровень канала данных в СПД АТМ
Рассмотрим эталонную модель СПД АТМ, которая состоит из трех уровней: физического, АТМ-уровня и уровня адаптации. Поверх СПД АТМ пользователь может поместить, например, стек TCP/IP.
Физический уровень в АТМ определяет правила передачи и приема данных в форме потока битов и преобразования их в ячейки. Носителями этого потока могут быть разные физические среды, у АТМ здесь нет ограничений. Охватывает физический уровень и уровень канала данных в OSI.
АТМ-уровень отвечает за транспорт ячеек. Он определяет формат ячейки, заголовок, его содержимое, отвечает за установление и поддержание виртуальных соединений. Управление потоком и перегрузками также расположено здесь.
Уровень адаптации обеспечивает приложениям-пользователям возможность работы в терминах пакетов или подобных им единиц, а не ячеек.
Поскольку физический уровень АТМ на подуровне физической зависимости не предъявляет каких-то специальных требований к физической среде, то сосредоточим внимание на ТС-подуровне, т.е. на подуровне подготовки ячеек.
Передача ячеек.
1. Вычисление контрольной суммы заголовка. Заголовок состоит из 5 байт, четыре из которых идентифицируют виртуальное соединение и несут контрольную информацию, а один — содержит контрольную сумму. Контрольная сумма защищает только первые четыре байта и не затрагивает данные в ячейке. Контрольная сумма вычисляется как CRC-код, т.е. как остаток от деления содержимого четырех байтов на полином $x8 + \boldsymbol{x2} + \boldsymbol{x} + 1$. К этому остатку добавляется константа 01010101 для повышения надежности в случае, если заголовок содержит много нулей. Поскольку контрольная сумма защищает только заголовок, то этот байт называется HEC (Header Error Control — контроль ошибки в заголовке).
Для надежной передачи ячеек была предложена схема, в которой две последовательные ячейки объединяются через EXCLUSIVE OR, после чего получается новая ячейка, добавляемая в последовательность после первых двух. В результате ячейку, принятую с ошибкой или потерянную, легко можно восстановить. После того как HEC вычислен и добавлен в заголовок, ячейка готова к передаче.
Среда передачи может быть как синхронной, так и асинхронной. В асинхронной среде ячейка посылается сразу, как только она готова к передаче. В синхронной среде ячейка передается в соответствии с временными соглашениями. Если ячейки для передачи нет, то ТС-подуровень должен сгенерировать специальную ячейку ожидания.
Еще один вид служебных ячеек на ТС-подуровне — OAM (Operation And Maintenance). Эти ячейки используются АТМ-переключателями для проверки работоспособности системы. Ячейки ожидания обрабатываются ТС-подуровнем, а OAM-ячейки передаются на АТМ-уровень.
Важной функцией ТС-подуровня является также генерирование ячеек в формате физической среды передачи. Это означает, что ТС- подуровень генерирует обычную АТМ-ячейку и упаковывает ее в кадр надлежащей среды передачи.
Прием ячеек. На выходе ТС-подуровень формирует HEC- заголовок, преобразует ячейку в кадр, формирует АТМ-ячейки и передает поток битов на физический уровень. На противоположном конце соединение ТС-подуровень производит те же самые действия, но в обратном порядке: разбивает поток битов на кадры, выделяет ячейки, проверяет HEC-заголовки и передает ячейки на АТМ- уровень.
Как определить границы кадра? На ТС-подуровне имеется сдвиговый регистр на 40 бит. Если из этих 40 бит первые 8 бит представляют собой HEC, то последующие 32 бит — это заголовок ячейки. Если это условие не выполнено, то все сдвигается на один бит и проверка повторяется. Этот процесс продолжается до тех пор, пока не будет обнаружен HEC. Представленная схема распознавания заголовка кадра ненадежна.

Билет № 27.
Спутниковые системы связи: организация, классификация и сравнительный анализ классов (примеры).

Идея создания системы связи на основе отражающего объекта, расположенного высоко над землей, давно витала в головах исследователей. Ее привлекательность состояла в том, что чем выше отражающий объект расположен над землей, тем большую часть поверхности Земли можно охватить связью при одинаковом угле обзора.
Сначала пытались использовать в качестве такого объекта металлизированный воздушный шар, воздушные плотные массы и т.д. Однако сигнал возвращался настолько слабым, что практическое использование такой системы было невозможно. Первый спутник связи был запущен в СССР в 1962 г. Основное его отличие от всего предложенного ранее заключалось в том, что он усиливал сигнал прежде чем отправить его назад на Землю. Спутник связи имеет несколько приемопередатчиков — транспондеров. Каждый транспондер слушает свою часть спектра, усиливает полученный сигнал и передает его обратно на Землю в требуемом направлении, на требуемой частоте, отличной от частоты приема, во избежание интерференции с принимаемым сигналом. Луч транспондера может быть по желанию либо широким, покрывающим большую территорию, либо, наоборот, узконаправленным.
СПД на основе спутниковых систем связи (СЗ) имеют существенные отличия от наземных кабельных СПД. Несмотря на то что сигнал распространяется со скоростью света, вследствие больших расстояний задержка при его передаче составляет 250... 300 мс в отличие от 3...5 мкс/км в кабельных системах (в коаксиальном кабеле, оптоволоконном и т. д.).
Спутниковые системы принципиально вещательного типа, что для некоторых приложений очень важно. При этом стоимость передачи не зависит от того, скольким получателям сообщение предназначено. Однако проблема безопасности передаваемой информации при этом требует особого внимания, так как все слышат все, что передается, а следовательно, необходимо шифрование. Стоимость передачи также не зависит от расстояния. Для справки: средняя стоимость спутниковой платформы оценивается сейчас в 150... 210 млн долл. США, к чему необходимо добавить стоимость приемопередающей аппаратуры (транспондеров) и запуска, т.е. еще 150 млн долл.
По высоте и форме орбиты все СЗ подразделяются на

- геостационарные (GEO — Geostationary Earth Orbit),
- высокоэллиптические (HEO — High Elliptic Orbit),
- средне-орбитальные (MEO — Middle Earth Orbit)
- низкоорбитальные (LEO — Low Earth Orbit).

Билет 28.
Спутниковые системы связи: геостационарные и низкоорбитальные спутниковые системы (примеры).

Геостационарные спутники

Для геостационарных спутниковых систем связи (ГСЗ) характерны:

- большая область покрытия;
- очень большая задержка распространения сигнала;
- высокая стоимость;
- большая мощность приемопередающей аппаратуры;
- покрытие больших широт
- стационарное размещение в пространстве относительно Земли.

Говоря о геостационарных спутниках надо также иметь в виду, что на высоте 1500 ... 5000 км и 13000...20000 км находятся радиационные пояса Ван Алена, которые препятствуют распространению радиосигналов и требуют дополнительной мощности передатчика для их преодоления.
Серьезную техническую проблему для ГС3 долгое время представляло создание наземных терминалов. Из-за относительно малой мощности принимаемого сигнала наземные терминалы оборудовались большими антеннами (диаметром от 5 м и больше). Ориентация в пространстве на спутник таких массивных инженерных сооружений представляла собой непростую задачу. Относительно новой технологией для $С^3$ является технология малых антенн, называемых VSAT (Very Small Aperture Terminals — терминалы с очень маленькой апертурой), т.е. антенн с маленьким радиусом. Такой терминал имеет антенну диаметром 0,5...2,5 м, способную излучать сигнал мощностью 1 Вт, и может передавать данные со скоростью примерно 19,2 Кбит/с, а принимать — Со скоростью 512 Кбит/с.

Низкоорбитальные спутники

низкоорбитальные, или LEO-спутники, двигаются на высотах от 500 до 1500 км. Ниже 500 км спутники не опускают, поскольку в этом случае их начинает тормозить атмосфера Земли (хотя и очень разреженная) и они начинают намного интенсивнее сваливаться к Земле. Малый период обращения (в среднем они делают один оборот вокруг Земли за 100... 120 мин) сокращает время пребывания спутника в пределах видимости из одной точки поверхности до 10... 15 мин.
Существенно лучшие энергетические характеристики LEO-спутников позволяют уменьшить наземные терминалы до размеров обычного сотового телефона или радиомодема. Однако, для обеспечения постоянной связи требуются группировки из нескольких десятков LEO-спутников, цепочками вращающихся на разнесенных орбитах, при этом чем ниже орбита, тем больше требуется спутников и орбит, причем важно точно поддерживать взаимное расположение спутников.
LEO-спутники проигрывают своим более высоким конкурентам во времени жизни из-за частого переключения энергетической системы с солнечных батарей на аккумуляторы и соответственно из-за увеличения числа циклов зарядки-разрядки батарей, вследствие чего срок их службы составляет 5... 8 лет.
Обычно LEO-спутник несет один трапспондер, реже два или три, один из которых резервный. Однако по сравнению с GEO-спутниками на них чаще устанавливается сложная электроная аппаратура коммутации потоков в целях осуществления интеллектуальной межспутниковой связи с четырьмя соседями, при этом число необходимых станций может быть небольшим.
Первый проект использования LEO-спутников Иридиум выдвинула компания Моторола в 1990 г. Идея была очень проста: когда пятно луча одного спутника уходило из определенного места, к этому месту подлетал другой спутник, пятно которого охватывало это место. Подлетевший спутник подхватывал передачу-прием, которую вел улетающий спутник, и связь сохранялась.
Основной целью этого проекта являлось обеспечение связи с наземными средствами (даже портативными) по всей поверхности Земли.
Итак, для низкоорбитальных систем характерны:

- малая область покрытия;
- малая задержка сигнала и низкая стоимость;
- использование маломощного сигнала;
- высокая наземная скорость спутника;
- короткое время жизни спутника (по сравнению с геостационарными)

Билет 29.
Спутниковые системы связи: VSAT спутниковые системы и спутниковые системы для персонального использования (примеры).

Персональная спутниковая связь

Традиционно инфраструктура связи России строилась на кабельных системах, и в настоящее время около 100 тыс. населенных пунктов нашей страны не имеют никакой оперативной связи с внешним миром, поскольку таковая до них еще ≪не дотянулась. Географически эти населенные пункты сосредоточены на севере страны, в Сибири и на Дальнем Востоке. В наши дни связь с этими районами можно обеспечить с помощью $С^3$, не требующих в отличие от кабельной связи огромных средств на развертывание наземной инфраструктуры. Огромные пространства и низкая плотность населения на большей части территории нашей страны делают наземные каналы связи экономически неэффективными. Несмотря на бурный рост сотовых сетей связи различных стандартов, услуги персональной спутниковой связи в этих районах все еще предпочтительнее с финансовой точки зрения. Понятие персональной спутниковой связи (как в мире, так и в России) с самого начала означало не чисто спутниковую связь, а комбинацию С3 с существующими сотовыми сетями. При этом основным назначением спутниковой связи является дополнение и расширение возможностей сотовой связи за пределами ее зон покрытия, где создание инфраструктуры других видов связи по экономическим либо технологическим причинам нецелесообразно. Многорежимные абонентские терминалы при работе в зонах сотовой связи автоматически устанавливают соединение с сотовой сетью (с использованием одного из стандартов: GSM, AMPS, CDMA), а за ее пределами используют спутниковый ретранслятор.
Стационарные спутниковые абонентские терминалы особенно выгодно применять в тех районах, где связь отсутствует вообще, поскольку они обеспечивают (через спутник) подключение к наземным сетям общего пользования (в том числе и телефонным — ТфОП).
Принцип организации персональной спутниковой связи достаточно прост. Если это возможно, терминал ищет наземную сотовую сеть и работает через нее, а если наземная сотовая сеть недоступна, переключается в спутниковый режим работы. Сигналы со спутников направляются на станции сопряжения, связанные с сетями общего пользования, а глобальное покрытие позволяет организовать телефонную связь между любыми населенными пунктами России. В настоящее время в нашей стране персональная телефонная связь возможна только посредством трех спутниковых систем связи: Inmarsat, Globalstar и ICO (Intermediate Circular Orbit), первая и последняя из которых основаны на использовании среднеорбитальных спутников, а вторая — низкоорбитальных.

VSAT-сети

В настоящее время VSAT-сети — наиболее динамично развивающаяся категория терминалов для $С^3$. Если в конце 1999 г. в мире было установлено более 300 тыс. приемопередающих терминалов VSAT, то к концу 2000 г. их уже стало уже около 500 тыс. Аналитики продолжают утверждать, что рынок VSAT еще далек от насыщения даже в таких развитых странах, как США, Великобритания и Япония.
Для многих крупных и средних предприятий с филиалами, разбросанными по всему миру, электронный документооборот и другие электронные формы ведения бизнеса стали необходимой потребностью. Как показывает мировой опыт, их требованиям в наибольшей степени отвечают телекоммуникационные услуги глобальных корпоративных сетей связи. Современные глобальные корпоративные сети чаше всего базируются на технологии VSAT, т. е. на использовании малогабаритных спутниковых терминалов и антенн диаметром 1,0...2,5 м.
Этот вид сетей широко распространен во многих странах, но особенно актуальны они в России, где наземная инфраструктура связи на значительной части территории неразвита по климатическим либо геологическим причинам. Оптимальным решением для труднодоступных районов считается сочетание магистральных каналов наземной связи и выделенных систем $С^3$. При этом наиболее рентабельными системы $С^3$ становятся там, где развертывание наземных сетей экономически нецелесообразно или просто невозможно.
Аналитики предсказывают рост индустрии VSAT по мере развития традиционных сфер ее применения — электронной торговли, банковских и биржевых операций, обеспечения телекоммуникационными услугами жителей труднодоступных районов. Они считают, что технология VSAT постепенно станет одной из господствующих в области связи.
Выделенные сети на базе VSAT-терминалов способны предоставить своим удаленным пользователям широкий спектр услуг, включая высококачественную телефонную и факсимильную связь, передачу данных с различной скоростью, организацию видеоконференций и распределение телепрограмм.

VSAT-сети телефонной и факсимильной связи могут иметь любую топологию: от простейшей двухточечной до полнодоступной схемы каждый с каждым. Выделение спутникового канала может быть организовано для постоянного использования или по требованию. При создании сетей корпоративной связи (т.е. СПД предприятия) в сельской местности или при подключении удаленных станций к существующим сетям, в том числе к коммутируемой сети общего пользования (например, телефонной), данный вид услуги является приоритетным. Современное VSAT-оборудование обеспечивает возможность подключения к наземным сетям ISDN. Типовая скорость передачи данных при таком соединении (с одним интерфейсом BRI) колеблется от 128 до 160 Кбит/с. Использование современных алгоритмов сжатия данных позволяет ≪упаковать≫ речевой канал в полосу пропускания 6,4 или 4,8 Кбит/с, благодаря чему пропускная способность спутникового канала при передаче речи повышается в 10—12 раз.
VSAT-терминалы поддерживают практически все типовые сетевые интерфейсы: RS232, RS449/422, Ethernet (ШЕЕ 802.3), Token Ring (IEEE 802.5), и поэтому могут использоваться для объединения локальных сетей на базе наиболее популярных протоколов IP, IPX, Net-BIOS. Кроме того, применение многопротокольной среды и технологии Frame Relay позволяет создавать сети с гибкой сменой скоростей и качества услуг передачи. Например, скорость передачи в этих сетях может изменяться от 64 Кбит/с до 8,448 Мбит/с. Основными потребителями таких услуг высокоскоростной передачи данных и мультимедиа являются банки и страховые компании, средства массовой информации, государственные учреждения.
Технология VSAT допускает также создание корпоративных многоцелевых сетей с коммутацией пакетов и большим числом удаленных станций. Скорость передачи в этих сетях обычно не превышает 64 Кбит/с, а передача данных осуществляется с использованием стандартных протоколов X.25, LAP-B, HDLC. Такие сети с множеством узлов характеризуются асимметричным трафиком с лавинообразной или непредсказуемой нагрузкой. Однако VSAT-технология позволяет организовать постоянный или дополнительный канал по требованию и обеспечить приоритезацию трафика. В качестве примеров можно привести сети бензозаправочных станций с проверкой кредитных карточек в режиме реального времени, сети контроля за банкоматами, сети сбора и обработки телеметрической и метеорологической информации и т.п.

Билет № 30.
Тенденции развития современных спутниковых систем связи.

Новым техническим решением является использование на GEO-спутниках антенн с десятками лучей. Как правило, лучи имеют ширину диаграммы направленности (ДН) 1 ...2° и обеспечивают плотное покрытие рабочей зоны. Для каждого луча выделен свой частотный ствол (стволы) ретранслятора. Смежные лучи развязаны по частоте, а несмежные лучи с совпадающими частотами должным образом поляризованы.
Для реализации антенных систем LEO-спутников также предусматривается многолучевая технология, имеющую в качестве основы фазированные антенные решетки. Такая антенна представляет собой решетку из отдельных антенн, диаграмму направленности каждой из которых можно формировать и управлять независимо с помощью компьютера.
По-видимому, в ближайшие несколько лет будут заявлены и новые сверхинформативные спутниковые системы. Уже сегодня активно идет процесс их системной интеграции.
Начало эксплуатации сверхинформативных систем позволит предоставить абонентам принципиально новые услуги связи, например видеотелефонную связь, формирование пакета телевизионных программ по заказу абонента и многое другое. Из этого конечно же не следует, что в XXI в. не будут развиваться глобальные, континентальные и национальные спутниковые системы, действующие в настоящее время. Однако уже в начале XXI в. постепенное насыщение рынка телекоммуникаций приводит к необходимости пересмотра их организационной структуры, стратегических планов развития и взаимной технической и коммерческой координации в целях оптимального участия в формировании и создании единого мирового информационного пространства.
Уже сейчас идет работа над Межпланетарным Интернетом. В мае 2009 г. на международной космической станции успешно прошли испытания нового стека протоколов, реализующего технологию сетей, адаптирующихся к задержкам — DTN (Delay Tolerant Networking). DTN-протоколы позволяют распространить наземный Интернет в космос. Они решают целый ряд серьезных проблем, например уменьшают большую задержку пакета при его перемещении по космическим каналам связи, изменчивость этой задержки вследствие изменения взаимного расположения узлов космической сети на орбитах и неравномерность сетевого трафика и высокого уровня помех в канале вследствие солнечной радиации.
По-видимому, в недалеком будущем развитие средств связи в целом приведет к иной форме восприятия мира и новому этапу развития цивилизации.

Билет № 31.

Проблемы передачи данных на канальном уровне (Сервис, предоставляемый сетевому уровню, Разбиение на кадры, Контроль ошибок, Управление потоком). Простейшие протоколы канала данных (Симплекс протокол без ограничений, Симплекс старт стопный протокол, Симплексный протокол для канала с шумом).

На уровне канала данных решается ряд проблем, присущих только этому уровню:

- реализация сервиса для сетевого уровня,
- объединение битов, поступающих с физического уровня, в кадры,
- обработка ошибок передачи,
- управление потоком кадров.

Основная задача канального уровня - обеспечить сервис сетевому уровню. Назначение этого сервиса - помочь передать данные от процесса, на сетевом уровне одной машины, процессу, на сетевой уровень другой машины.

Канальный уровень может обеспечивать различные классы сервиса. Однако, есть три общие класса сервиса:

1. Сервис без уведомления и без соединения.
2. Сервис с уведомлением и без соединения.
3. Сервис с уведомлением и с соединением.

Сервис без уведомления и без соединения не предполагает, что прием переданного кадра должен подтверждаться, что до начала передачи должно устанавливаться соединение, которое после передачи должно разрываться. Если в результате помех на физическом уровне кадр будет потерян, то никаких попыток на канальном уровне его восстановить не будет.

Следующий класс сервиса - уведомление без соединения. В этом классе получение каждого посланного кадра должно быть подтверждено. Если подтверждения не пришло в течение определенного времени, то кадр должен быть послан опять. Этот класс сервиса используется в ненадежной физической среде передачи, например, беспроводной.

Наиболее сложный класс сервиса на канальном уровне - сервис с уведомлением и соединением. Этот класс сервиса предполагает, что до начала передачи между машинами устанавливается соединение, и данные передаются по этому соединению. Каждый передаваемый кадр нумеруется и канальный уровень гарантирует, что кадр будет обязательно получен и только один раз, а все кадры будут получены в надлежащей последовательности. При сервисе без соединения этого гарантировать нельзя потому, что потеря уведомления о получении кадра приведет к его пересылке так, что может появиться несколько идентичных кадров.

При сервисе с уведомлением и соединением передача разбивается на три этапа. На первом этапе устанавливают соединение: на обеих машинах инициируют счетчики, отслеживающие какие кадры были приняты, а какие нет. На втором этапе передают один или несколько кадров. На третьем - соединение разрывают: переменные, счетчики, буфера и другие ресурсы, использованные для поддержки соединения, освобождаются.

Разбиение на кадры

Сервис, создаваемый канальным уровнем для сетевого, опирается на сервис, создаваемый физическим уровнем. На физическом уровне протекают потоки битов. Посланное количество битов не обязательно равно принятому, значение посланного бита так же не обязательно равно принятому. Поэтому нужны специальные усилия на канальном уровне по обнаружению и исправлению ошибок.

Типовой подход к решению этой проблемы - разбиение потока битов на кадры, подсчет контрольной суммы для каждого кадра при посылке данных. При приеме контрольная сумма вычисляется для каждого кадра заново и сравнивается с той, что храниться в кадре. Если они различаются, то это признак ошибки передачи. Канальный уровень должен принять меры к исправлению ошибки, например, сбросить плохой кадр, послать сообщение об ошибке тому, кто прислал этот кадр.

Один из способов разбиения потока битов на кадры - делать временную паузу между битами разных кадров. Однако, в сети, где нет единого таймера, нет гарантии, что эта пауза всегда будет одинаковой или, наоборот, не появятся новые паузы.

Другие методы:

1. счетчик символов. В начале каждого кадра указывается, сколько символов в кадре. При приеме число принятых символов подсчитывается опять. Этот метод имеет существенный недостаток: счетчик символов может быть искажен при передаче. Тогда принимающая сторона не сможет обнаружить границы кадра. Даже обнаружив не совпадение контрольных сумм, принимающая сторона не сможет сообщить передающей какой кадр надо переслать, сколько символов пропало. Этот метод ныне используется редко.
2. вставка специальных стартовых и конечных символов. Метод построен на вставке специальных символов. Обычно для этого используют последовательность DLE S*TX* для начала кадра и DLE *ETX* для конца кадра. При этом методе, если даже была потеряна граница текущего кадра, можно найти границу следующего. Для этого надо просто искать ближайшую последовательность DLE S*TX* или DLE *ETX*. Здесь есть одна опасность: при передаче чисел или программы в объектном коде такие последовательности могут уже содержаться в передаваемых данных. Для решения этой проблемы используется прием экранирования: каждая последовательность DLE просто дублируется в передаваемых данных. Поэтому, если при приеме есть два последовательных DLE, то один удаляется. Основным недостатком этого метода является то, что он жестко связан с размером байта и кодировкой ASCII.
3. вставка стартовых и концевых битов. Метод состоит в том, что каждый кадр начинается и заканчивается специальным флаг-байтом: 01111110. Посылающая сторона, встретив последовательно 5 единиц, обязательно вставит 0. Принимающая сторона, приняв 5 последовательных единиц, обязательно удалит следующий за ними 0. Если в передаваемых данных встретиться конфигурация флаг-байта, то она будет преобразована в конфигурацию 011111010.
4. комбинация этих методов. Например, счетчик символов с одним из выше перечисленных методов. Тогда, если число символов в кадре совпадает с кодировкой границы кадра, кадр считается переданным правильно.

Обнаружение ошибок

Если нам нужен сервис с подтверждением и с соединением. Для решения этой проблемы устанавливают обратную связь между отправителем и получателем в виде кадра подтверждения. Если кадр-подтверждение несет положительную информацию, то считается, что переданные кадры прошли нормально, если там сообщение об ошибке, то переданные кадры надо передать заново. Однако, возможны случаи когда из-за ошибок в канале кадр исчезнет целиком.

Для решения этой проблемы на канальном уровне вводят таймеры. Таймер это счетчик, который увеличивает или уменьшает свое значение на единицу автоматически, при получении тактирующего импульса. Если отправитель не получит подтверждение раньше, чем истечет время, установленное на таймере, то он будет считать, что кадр потерян и повторит его еще раз.

Однако если кадр подтверждение был утерян, то вполне возможно, что один и тот же кадр получатель получит дважды. Для решения этой проблемы каждому кадру присваивают порядковый номер. С помощью этого номера получатель может обнаружить дубли.

Управление потоком

Другой важной проблемой, которая решается на канальном уровне является управление потоком. Суть этой проблемы в том, что отправитель будет слать кадры столь часто, что получатель не будет успевать их обрабатывать.

Для борьбы с такими ситуациями вводят управление потоком. Это управление предполагает обратную связь между отправителем и получателем, которая позволяет им урегулировать такие ситуации. Есть много схем управления потоком, но все они в основе своей используют следующий сценарий. Прежде чем отправитель начнет передачу он спрашивает у получателя, сколько кадров тот может принять. Получатель сообщает ему определенное число кадров. Отправитель, после того как передаст это число кадров, должен приостановить передачу и спросить получателя опять как много кадров тот может принять.

Симплекс протокол без ограничений

Данные передаются только в одном направлении. Получатель и отправитель всегда готовы к отправке и получению данных. Время обработки данных игнорируется. Предполагается, что буфер неограниченного размера. Ну, и, наконец, данные в канале не теряются и не искажаются.

Симплекс старт стопный протокол

Тоже самое, но без условия обработки поступающих данные сколь угодно быстро. Остальное выполнено: канал абсолютно надежный, трафик однонаправленный.

Основная проблема здесь как предотвратить ситуацию, когда отправитель "заваливает" данными получателя. Решением такой проблемы может быть введение специальных, коротких служебных сообщений. Получатель, получив один или несколько кадров, отправляет отправителю специальный, короткий кадр, означающий, что отправитель может передавать следующий. Это, так называемый, старт-стопный протокол

Симплексный протокол для канала с шумом

Основная проблема здесь состоит в том, что кадр с подтверждением о получении может потеряться целиком. Поскольку проблема различения стоит для кадров m и m+1, то достаточно одного разряда. 0 для только что посланного кадра и 1 – для кадра, посланного повторно. Все кадры, не содержащие корректной нумерации, просто сбрасываются при приеме.

Билет № 32.

Проблемы передачи данных на канальном уровне (Сервис, предоставляемый сетевому уровню, Разбиение на кадры, Контроль ошибок, Управление потоком). Обнаружение и исправление ошибок (Коды исправляющие ошибки, Коды обнаруживающие ошибки).

Обнаружение и исправление ошибок

В разных средах характер ошибок разный. Ошибки могут быть одиночные, а могут возникать группами, сразу несколько. Недостатком групповых ошибок является то, что их труднее обнаруживать и исправлять, чем одиночные.

Коды с исправлением ошибок.

Для надежной передачи кодов было предложено два основных метода. Первый - внести избыточность в форме дополнительных битов в передаваемый блок данных так, чтобы, анализируя полученный блок, можно было бы указать, где и какие возникли искажения. Это, так называемые, **коды с исправлением ошибок.** Второй - внести избыточность, но лишь настолько, чтобы, анализируя полученные данные, можно было сказать, есть в переданном блоке ошибки или нет. Это, так называемые, **коды с обнаружением ошибок.**

Пусть данные занимают *m* разрядов и мы добавляем *r* разрядов, как контрольные разряды. Нам надо передать слово длины n (n=m+r), которое называют ***n*-битовым кодословом.**

Количество разных битов в двух кодословах называется **расстоянием Хемминга.**

Если мы хоти обнаруживать *d* ошибок, то надо чтобы кодослова отстояли друг от друга на расстояние *d*+1.

Если мы хотим исправлять ошибки, то надо чтобы кодослова отстояли друг от друга на 2*d*+1.

Простым примером кода с обнаружением одной ошибки является код с битом четности.

Оценим минимальное количество контрольных разрядов, необходимое для исправления одиночных ошибок. Пусть у нас есть код из *m* бит сообщения и *r* контрольных бит. Каждое из 2^m правильных сообщений имеет *n* неправильных кодослов на расстоянии 1. Таким образом, с каждым из 2^m кодослов связано *n*+1 кодослов. Так как общее число кодослов - 2^n, то $(n+1)2^m \le 2^n$, учитывая что

$n = m + r$ получаем $(m+r+1) \le 2^r$.

Для заданного *m*, эта формула дает предельно минимальное число контрольных разрядов, необходимое для исправления единичных ошибок. Этот теоретический предел достижим при использовании метода, предложенного Хеммингом. Идея его в следующем: все биты, номера которых есть степень 2 (1,2,4,8,16 и т.д.) - контрольные, остальные - биты сообщения. Каждый контрольный бит отвечает за четность группы битов, включая себя. Один и тот же бит может относиться к разным группам. Значение бита сообщения определяется по значениям контрольных битов. Чтобы определить какие контрольные биты контролируют бит в позиции *k* надо представить значение *k* по степеням двойки. Получив кодослово, получатель устанавливает специальный счетчик в ноль. Затем он проверят каждый контрольный бит на предмет правильности четности. Если четность нарушена, то порядковый номер этого бита заноситься в счетчик. Если после этой проверки счетчик ноль, то все в порядке. Если нет, то он содержит номер неправильного разряда. Код Хемминга может исправлять только одиночные ошибки. Однако есть прием, который позволяет распространить идеи Хемминга на случай групповых ошибок.

Коды обнаруживающие ошибки

Применение техники четности "в лоб" в случае групповых ошибок не даст нужного результата. Однако ее можно скорректировать. Пусть нам надо передать *n* слов по *k* бит. Расположим их в виде матрицы *n x k*. Для каждого из *n* столбцов вычислим бит четности и разместим их в дополнительной строке. Получившаяся матрица затем передается по строкам. По получению матрица восстанавливается и если нарушен хоть один бит, то весь блок передается повторно. Этот метод позволяет обнаружить одиночный пакет ошибок длины *n*. Против групповых ошибок длины *n*+1 он бессилен.

Поэтому на практике применяют другую технику, которая называется полиномиальным кодом или циклическим избыточным кодом (Cyclic Redundancy Code) или CRC кодом.

CRC коды построены на рассмотрении битовой строки как строки коэффициентов полинома. *k* битовая строка - коэффициенты полинома степени *k*-1. Самый левый бит строки - коэффициент при старшей степени. Например, строка 110001 представляет полином $x^5+x^4+x^0$.

Полиномиальная арифметика выполняется по модулю 2, т.е. сложение и вычитание происходят без переноса разрядов. Так, что обе это операции эквивалентны EXCLUSIVE OR. Например,

```
  10011011        00110011        11110000        01010101
 +11001010      + 11001101      - 10100110      - 10101111
 ----------      ----------      ----------      ----------
  01010001        11111110        01010110        11111010
```

Деление выполняется как обычно в двоичной системе с той лишь разницей, что вычитание выполняется по модулю два.
Использование полиномиальных кодов при передаче заключается в следующем. Отправитель и получатель заранее договариваются о конкретном генераторе полиномов $G(x)$, у которого коэффициенты при старшем члене и при младшем члене должны быть равны 1. Пусть степень $G(x)$ равна r. Для вычисления контрольной суммы блока из m бит надо, чтобы обязательно $m>r$. Идея состоит в том, чтобы добавить контрольную сумму к передаваемому блоку, рассматриваемому как полином $M(x)$ так, чтобы передаваемый блок с контрольной суммой был кратен $G(x)$. Когда получатель получает блок с контрольной суммой, он делит его на $G(x)$. Если есть остаток, то были ошибки при передаче.
Алгоритм вычисления контрольной суммы:
Здесь r степень $G(x)$

1. Добавить r нулей в конец блока так, что он теперь содержит $m+r$ разрядов и соответствует полиному $x^r M(x)$;
2. Разделить по модулю 2 полином $x^r M(x)$ на $G(x)$;
3. Вычесть по модулю 2 остаток (длина которого всегда не более r разрядов) из строки, соответствующей $x^r M(x)$. Результат и есть блок с контрольной суммой (назовем его $T(x)$).

Этот метод позволяет отлавливать одиночные ошибки. Групповые ошибки длины не более r. Нечетное число отдельных ошибок. Идея обоснования этих утверждений состоит в следующем. Ошибки при передаче означают в терминах полиномов, что мы получим после передачи не $T(x)$, а $T(x) + E(x)$. Если степень $E(x)$ меньше степени $G(x)$, то остаток от деления никогда не будет равен 0. Степень, количество и вид термов в $G(x)$ определяет вид выявляемых ошибок этим методом.

Существует три международных стандарта на вид $G(x)$:

CRC-12 $= x^{12}+x^{11}+x^3+x^2+x+1$

CRC-16 $= x^{16}+x^{15}+x^2+1$

CRC-CCITT $= x^{16}+x^{12}+x^5+1$

CRC-12 используется для передачи символов из 6 разрядов. Два остальных -для 8 разрядных. CRC-16 и CRC-CCITT ловят одиночные, двойные ошибки, одиночные пакеты ошибок длины не более 16 и нечетное число изолированных ошибок с вероятностью 99,997%.

Билет № 33.

Проблемы передачи данных на канальном уровне (Сервис, предоставляемый сетевому уровню, Разбиение на кадры, Контроль ошибок, Управление потоком). Протоколы скользящего окна.

Для передачи в обоих направлениях можно потребовать на физическом уровне двух сиплексных каналов. Один для передачи кадров, другой - для передачи подтверждений. Можно смешивать трафики: кадры с данными и кадры с подтверждениями на одном канале. Получатель не сразу отправляет подтверждение, а ожидает от сетевого уровня очередного пакета. Как только такой пакет возникает, то канальный уровень помещает информацию о подтверждении в поле *ack* кадра с пакетом. Такой прием позволяет полнее использовать имеющуюся пропускную способность канала. Меньше кадров - меньше прерываний канального уровня на их обработку, меньше затрат на буферизацию. На канальном уровне должен быть фиксированный интервал времени, в течение которого канальный уровень ждет от сетевого попутного кадра. Если до истечения этого срока пакет не поступил, то канальный уровень отправляет подтверждение отдельным кадром.

Протоколы этого класса кроме вышеперечисленных проблем решают еще одну. У отправителя и получателя есть определенная константа *n* -число кадров, которые отправитель может послать без ожидания подтверждения каждого. По мере получения подтверждений, отправленные кадры будут сбрасываться из буфера отправителя, и буфер будет пополняться новыми.

Как только от сетевого уровня поступил еще один пакет ему присваивается первый свободный наибольший номер и верхняя граница окна отправителя поднимается. Как только приходит подтверждение, нижняя граница окна поднимается. Таким образом, в окне все время находятся неподтвержденные кадры.

Протокол скользящего окна в 1 бит

Машина, инициирующая обмен, берет пакет от сетевого уровня, формирует кадр и посылает его. Когда этот (или любой другой кадр) поступает, канальный уровень-получатель проверяет: не является ли этот кадр дубликатом. Если поступивший кадр тот, что ожидался, то он передается на сетевой уровень и окно получателя сдвигается вверх.

Поле уведомления содержит номер последнего кадра, полученного без ошибок. Если этот номер согласуется с номер кадра, который уровень-отправитель старается послать, то он считает, что кадр, хранящейся в буфере послан и сбрасывает его оттуда, забирая новый с сетевого уровня. Если номера не согласуются, то отправитель старается послать тот же кадр еще раз. В любом случае получив кадр, отправляют кадр.

Протокол с возвратом на n кадров и протокол с выборочным повтором

разрешается отправителю отправлять до *w* кадров, не ожидая их подтверждения. Надлежащим выбором значения *w* отправитель может заполнить все время, необходимое на оправку кадра и получение его подтверждения.

Эта техника известна как конвейер. Ее применение в случае ненадежного канала наталкивается на ряд проблем. Первая - что делать, если в середине потока пропадет или попадется поврежденный кадр? Получатель уже получит большое количество кадров к тому моменту, когда отправитель обнаружит, что что-то произошло. Когда получатель получил поврежденный кадр, он его должен сбросить, что делать с последующими кадрами? Помните, что канальный уровень обязан передавать пакета на сетевой уровень в том порядке, в каком их отправлял отправитель.

Есть два приема для решения этих вопросов: **откат** и **выборочный повтор**. При откате все кадры, поступившие после поврежденного, сбрасываются и не подтверждаются.

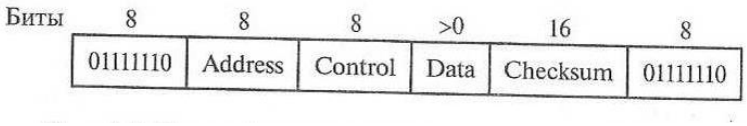
При выборочном повторе у получателя длина окна, как и у отправителя. Отправитель отмечает не подтвержденный кадр и посылает его повторно. Получатель не передает на сетевой уровень последовательность пакетов, если в ней есть разрывы.

Билет № 34.

Проблемы передачи данных на канальном уровне (Сервис, предоставляемый сетевому уровню, Разбиение на кадры, Контроль ошибок, Управление потоком). Пример протокола канального уровня (HDLC).

Рассмотрим группу протоколов имеющих одного предшественника - SDLC - Synchronous Data Link Control – протокол управления синхронным каналом, предложенным фирмой IBM в рамках SNA. ISO модифицировало этот протокол и выпустило под название HDLC -High level Data Link Control. МКТТ модифицировало HDLC для *X*.25 и выпустило под именем LAP - Link Access Procedure. Позднее он был модифицирован в LAPB.

Все они используют технику вставки специальных последовательностей битов, и являются бит – ориентированными протоколами.



Поле Control используется для последовательных номеров кадров, подтверждений и других нужд.

Поле Data может быть сколь угодно большим и используется для передачи данных.

Поле Checksum - это поле используется CRC кодом.

Флаговые последовательности 01111110 используются для разделения кадров и постоянно передаются по незанятой линии в ожидании кадра. Существуют три вида кадров: Information, Supervisory, Unnumbered.

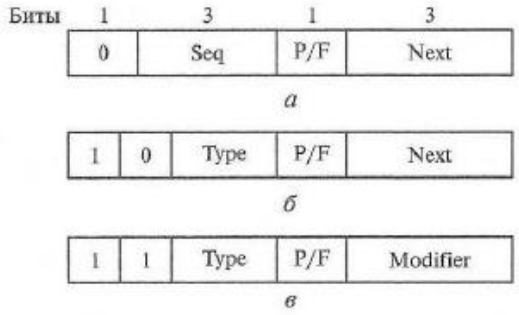


Рис. 4.6. Поле Control для кадров Information (*а*), Supervisory (*б*) и Unnumbered (*в*)

Как видно из размера поля Seq в окне отправителя может быть до 7 неподтвержденных кадров. Поле Next используется для посылки подтверждения вместе с передаваемым кадром. Подтверждение может быть в форме номера последнего правильно переданного кадра, а может быть в форме первого, не переданного кадра. Какой вариант будет использован - это параметр протокола.

Разряд *P/F* использует при работе с группой терминалов. Когда компьютер приглашает терминал к передаче, он устанавливает этот разряд в *P*. Все кадры, посылаемые терминалами, имеют здесь *P*. Если это последний кадр, посылаемый терминалом, то здесь стоит *F*.

Supervisory кадры имеют четыре типа кадров. Тип 0 - уведомление в ожидании следующего кадра (RECEIVE READY). Этот тип кадров используется, когда нет встречного трафика, чтобы передать уведомление в кадре с данными. Тип 1 - негативное уведомление (REJECT) - указывает на ошибку при передаче. Поле Next указывает номер кадра, начиная с которого надо пере послать кадры.

Тип 2 - RECEIVE NOT READY. Подтверждает все кадры, кроме указанного в Next. Используется, чтобы сообщить источнику кадров о необходимости приостановить передачу в силу каких-то проблем у получателя. После устранения этих проблем получатель шлет RECEIVE REDAY, REJECT или другой надлежащий управляющий кадр.

Тип 3 - SELECTIVE REJECT - указывает на необходимость пере послать только кадр, указанный в Next. LAPB и SDLC не используют этого типа кадров.

Третий класс кадров - Unnubered. Кадры этого класса иногда используются для целей управления, но чаще для передачи данных при ненадежной передаче без соединения.

Все протоколы имеют команду DISConnect для указания о разрыве соединения. SNRM и SABM - для установки счетчиков кадров в ноль, сброса соединения с начальное состояние, установки соподчиненности на линии. Команда FRMR - указывает на повреждение управляющего кадра (например, контрольная сумма верная, а вот значения полей противоречивы).

Билет № 35.
Архитектура Frame Relay и ее канальный уровень.

Ретрансляция кадров (Frame Relay — FR) — это СПД с коммутацией пакетов для сетей класса WAN, т.е. для сетей, объединяющих несколько ЛВС.

Архитектура Frame Relay

Технология Frame Relay использует для передачи данных технику виртуальных соединений, аналогичную той, которая применяется в СПД ATM, однако стек протоколов FR передает кадры (при установленном виртуальном соединении) по протоколам только физического и канального уровней, а третий пакетный уровень здесь не используется. При таком подходе уменьшаются накладные расходы на передачу пакетов локальных сетей, так как эти пакеты вкладываются сразу в кадры канального уровня СПД FR.
Протокол канального уровня FR использует кадры переменной длины и работает только по виртуальным соединениям, которые могут быть постоянными (PVC) или коммутируемыми (SVC). Прежде чем два узла начнут обмениваться информацией, между ними необходимо установить виртуальное соединение.
Постоянное виртуальное соединение PVC (Permanent Virtual Circuits) устанавливается между двумя узлами вручную в процессе конфигурирования сети, т. е. когда в каждом узле коммутации вручную прописывают, на какой выход передавать кадры определенного виртуального соединения. Для этого пользователь сообщает провайдеру FR-услуг или сетевому администратору, какие узлы должны быть соединены, и последний устанавливает PVC между этими узлами.
PVC включает в себя конечные станции, среду передачи и все коммутаторы, расположенные между конечными станциями. После установки для PVC резервируется определенная часть полосы пропускания каналов, поэтому двум конечным станциям не требуется каждый раз устанавливать или сбрасывать соединение. При этом благодаря методу статистического мультиплексирования несколько PVC могут разделять разные полосы одного канала передачи.
Статистическое мультиплексирование — это метод мультиплексирования, при котором полоса пропускания канала распределяется между потоками данных (виртуальными каналами) по мере необходимости.
Основной целью статистического мультиплексирования является кодирование с постоянным качеством за счет выделения большей полосы более сложным динамичным потокам.
Коммутируемые виртуальные соединения — SVC (Switched Virtual Circuits) устанавливаются по мере необходимости, т.е. всякий раз, когда один узел пытается передать данные другому узлу, и динамически, как в телефонных сетях, на основе информации, заложенной при инициализации СПД.
PVC имеют два преимущества по сравнению с SVC:

- сеть, в которой используются SVC, должна тратить время на установление соединений, а PVC устанавливаются предварительно, а значит, обеспечивают более высокую производительность.
- PVC обеспечивают лучший контроль над СПД, так как провайдер или сетевой администратор может заранее выбирать маршрут, по которому будут передаваться кадры.

Однако и SVC имеют ряд преимуществ по сравнению с PVC:

- Используют полосу пропускания только по мере необходимости, тогда как PVC должны постоянно ее резервировать на случай, если она понадобится;
- требуют меньшей административной работы, поскольку устанавливаются автоматически, а не вручную;
- обеспечивают отказоустойчивость, т.е. когда коммутатор, находящийся на пути соединения, выходит из строя, другие коммутаторы выбирают альтернативный путь.

Канальный уровень FR

На канальном уровне СПД FR используется бит-ориентированный синхронный протокол LAP-F (стандарт Q.922 МСЭ), являющийся весьма упрощенной версией протокола LAP-D, заимствованного из технологии ISDN, который, в свою очередь, является упрощенной версией протокола.
У протокола канального уровня LAP-F есть два режима работы: основной (core) и управляющий (control). В основном режиме кадры передаются без преобразования и контроля, как и в коммутаторах локальных сетей. За счет этого СПД FR обладают весьма высокой производительностью, так как кадры в коммутаторах не подвергаются преобразованию, а сеть не передает подтверждения между коммутаторами на каждый пользовательский кадр. Пульсации трафика передаются СПД FR достаточно быстро и без больших задержек.

Протокол LAP-F работает на любых каналах сети ISDN, а также на каналах типа TI/EI. Терминальное оборудование посылает в сеть кадры LAP-F в любой момент времени, считая, что виртуальный канал в сети коммутаторов уже проложен. При использовании PVC оборудованию СПД PR требуется поддерживать только протокол LAP-F core.
Для установки SVC используется канал D пользовательского интерфейса, на котором по-прежнему работает протокол LAP-D, используемый для надежной передачи кадров в СПД ISDN. Устанавливают виртуальное соединение на основе адресов конечных абонентов, а также номера виртуального соединения, который в технологии Frame Relay называется Data Link Connection Identifier - DLCL
После установки коммутируемого виртуального канала с помощью протоколов LAP-D кадры могут транслироваться по протоколу LAP-F, который коммутирует их с помощью таблиц коммутации портов, где используются локальные значения DLCI. Протокол LAP-F в основном режиме работы выполняет не все функции канального уровня по сравнению с протоколом LAP-D.
Действие СПД FR заканчивается на канальном уровне, поэтому она хорошо согласуется с идеей тунелирования, т.е. инкапсуляции пакетов единого сетевого протокола, например IP, в кадры канального уровня любых сетей, составляющих Интернет. Процедуры взаимодействия протоколов сетевого уровня с протоколом канального уровня FR стандартизованы, например принята спецификация RFC 1490, определяющая методы инкапсуляции в трафик FR трафика сетевых протоколов и протоколов канального уровня локальных сетей.
Другой особенностью технологии FR является отказ от коррекции обнаруженных в кадрах искажений. Протокол LAP-F подразумевает, что конечные узлы будут обнаруживать и корректировать ошибки за счет работы протоколов транспортного или более высоких уровней. Это требует некоторой степени интеллектуальности от конечного оборудования, что по большей части справедливо для современных локальных сетей. В этом отношении технология FR близка к технологиям локальных сетей, таким как Ethernet, Token Ring и FDDI, которые тоже только отбрасывают искаженные кадры, а повторной их передачей не занимаются. Структура кадра протокола FR (LAP-F) (рис. 5.15) включает в себя следующие элементы:

1. Флаг — комбинация 01111110, которой начинаются и заканчиваются все кадры;
2. Заголовок (стандарты ANSI и МСЭ допускают размер заголовка до 4 байт), который содержит:

- адрес в пределах кадра FR (стандарт FRF), который занимает 6 бит первого байта и 4 бит второго байта заголовка кадра. Эти 10 бит представляют собой идентификатор виртуального канала передачи данных (Data Link Connection Identifier — DLCI) и определяют абонентский адрес в СПД FR;
- бит CF1, зарезервированный для возможного применения в различных протоколах более высоких уровней управления OSI. Этот бит не используется протоколом FR и прозрачно пропускается аппаратно-программными средствами СПД FR;
- бит расширения адреса (Extended Address — EA). Минимальная длина идентификатора канала передачи данных DLCI составляет 10 бит, входящих в два байта заголовка. Однако возможно расширение заголовка на целое число дополнительных байтов для указания адреса, состоящего более чем из 10 бит. Адрес может иметь длину 16 бит либо 23 бит. Для расширения используется бит EA в конце каждого байта заголовка; если он имеет значение 1, то это означает, что данный байт в заголовке последний. Стандарт FR рекомендует использовать заголовки, состоящие из двух байтов. В этом случае значение бита EA первого байта будет соответствовать 0, а второго — 1;
- бит уведомления (сигнализации) приемника о явной перегрузке (Forward Explicit Congestion Notification — FECN), устанавливающийся в **1,** если надо информировать получателя о том, что произошла перегрузка в направлении передачи данного кадра, т.е. что в этом направлении возник слишком большой поток кадров, в результате чего поток превзошел пропускную способность канала, выделенного под данное виртуальное соединение, и какие-то кадры были сброшены;
- бит уведомления (сигнализации) отправителя о явной перегрузке (Backward Explicit Congestion Notification — BECN), который устанавливается в 1 для уведомления отправителя сообщения о том, что произошла перегрузка в направлении, обратном направлению передачи содержащего этот бит кадра. Бит BECN может не использоваться терминалами абонентов;
- бит разрешения сброса (Discard Eligibility — DE), указывающий на то, что в случае перегрузки данный кадр может быть уничтожен в первую очередь, т.е. пользователю предоставлено право выбирать, какими кадрами он может пожертвовать. Однако при перегрузках узлы коммутации в СПД FR уничтожают не только кадры с битом DE.

3. Поле Информация, которое содержит данные пользователя и состоит из целого числа байтов. Его максимальный размер определен стандартом FR и равен 4 096 байт (минимальный размер — 1 байт). Содержание этого поля передается без изменений.
4. Поле Проверка, которое используется для обнаружения возможных ошибок при передаче и состоит из двух байтов. Это поле формируется с помощью CRC-кода аналогично протоколу HDLC.
Все указанные поля должны присутствовать в каждом кадре FR, передаваемом между двумя оконечными пользовательскими системами.__ Одним из основных отличий протокола FR от HDLC является то, что FR не предусматривает передачу управляющих сообщений (в нем нет командных или супервизорных кадров, как в HDLC). Для передачи служебной информации в FR используется специально выделенный канал D. Другим важным отличием FR является отсутствие нумерации последовательно передаваемых (принимаемых) кадров. Дело в том, что в протоколе FR нет механизмов для подтверждения правильно принятых кадров. Это означает, что этот протокол предполагает использование достаточно надежной физической среды. Протокол FR является весьма простым и включает в себя небольшой свод правил и процедур обмена данными. Основная процедура состоит в том, что если кадр получен без искажений, он должен быть направлен далее по соответствующему маршруту. При возникновении перегрузки узлы СПД FR могут сбрасывать любой кадр. Узлам в СПД FR разрешено уничтожать искаженные кадры, не уведомляя об этом пользователя. При этом искаженным считается кадр, у которого:

- нет корректного ограничения флагами;
- между флагами менее пяти байтов;
- присутствует ошибка контрольной суммы, хранящейся в двухбайтовом поле контрольной суммы FCS;
- искажено поле адреса (для случая, когда проверка не выявила ошибки в поле FCS);
- содержится несуществующий DLCI;
- превышен допустимый максимальный размер (в некоторых вариантах реализации стандартов FR возможна принудительная обработка кадров, превышающих допустимый максимальный размер).

Билет № 36.
Управление качеством сервиса и доступом в СПД Frame Relay.

Управление качеством сервиса

В технологии FR особое внимание уделено управлению качеством сервиса, предоставляемого транспортной среде. Вместо приоритезации трафика, как например в технологии FDD1, в этой технологии используется процедура заказа качества обслуживания при установлении соединения.
Для каждого виртуального соединения определяется несколько параметров, влияющих на качество обслуживания:

- CIR (Committed Information Rate) — согласованная скорость передачи данных, с которой сеть будет передавать данные пользователя;
- Be (Committed Burst Size) — согласованный объем пульсации, т.е. максимальное число байтов, которое сеть будет передавать от этого пользователя за фиксированный интервал времени T
- Be (Excess Burst Size) — дополнительный объем пульсации, т.е. максимальное число байтов, которое сеть будет пытаться передать сверх установленного значения параметра Bc за интервал времени T.

Если эти параметры определены, то время T= Bc/CIR. Если задать значения CIR и T, то можно найти значение всплеска трафика Bc. Гарантий по задержкам передачи кадров технология FR не дает, оставляя эту услугу сетям ATM.
Основным параметром, по которому абонент и сеть заключают соглашение при установлении виртуального соединения, является согласованная скорость передачи данных. Для постоянных виртуальных каналов это соглашение составляет часть контракта на пользование услугами СПД. При установлении коммутируемого виртуального канала заключение соглашения о качестве обслуживания является частью протокола LAP-D, в котором требуемые параметры CIR, Bc и Be передаются в пакете запроса на установление соединения. Так как скорость передачи данных измеряется в каком-то интервале времени, то интервал T и является таким контрольным интервалом, в котором проверяются условия соглашения. В обшем случае пользователь не должен за этот интервал времени передавать в сеть данные со средней скоростью, превосходящей CIR. Если же он нарушает соглашение, то СПД не только не гарантирует доставку кадра, но помечает этот кадр признаком DE (Discard Eligibility), равным 1, т.е. как кадр, подлежащий удалению. Однако кадры, отмеченные таким признаком, удаляются из СПД только в том случае, если коммутаторы СПД испытывают перегрузки. Если же перегрузок нет, то кадры с признаком DE = 1 доставляются адресату. Некоторые операторы СПД (поставщики услуг) предлагают значительные скидки при передаче кадров с битом DE, установленным в 1. Такое щадящее поведение сети соответствует случаю, когда общее количество данных, переданных пользователем в СПД за период T, не превышает объема Bc+ Be. Если же этот порог превышен, то кадр не помечается признаком DE, а немедленно удаляется из СПД.
Для контроля соглашения о параметрах качества обслуживания все коммутаторы СПД FR используют так называемый алгоритм дырявого ведра (Leaky Bucket). В этом алгоритме используется счетчик C поступивших от пользователя байтов, который каждые T секунд уменьшается на значение Bc (или же сбрасывается в 0, если значение счетчика меньше, чем Bc). Все кадры, данные которых не увеличили значение счетчика свыше порога Bc, пропускаются в сеть со значением признака DE = 0. Кадры, данные которых привели к значению счетчика, большему Bc, но меньшему Bc + Be, также передаются в сеть, но с признаком DE = 1. И наконец, кадры, которые привели к значению счетчика, большему Bc + Be, отбрасываются коммутатором.
Пользователь может договориться о включении не всех параметров качества обслуживания на данном виртуальном канале, а только некоторых. Например, можно использовать только параметры CTR и Bc. Этот вариант обеспечивает более качественное обслуживание, так как кадры никогда не отбрасываются коммутатором сразу. Коммутатор только помечает кадры, которые превышают порог Bc за время T, признаком DE = 1. Если в СПД не возникает перегрузок, то кадры такого канала всегда доходят до конечного узла, даже если пользователь постоянно нарушает договор СПД.
Популярен еще один вид заказа на качество обслуживания, при котором оговаривается только порог Be, а скорость CTR полагается равной нулю. Все кадры такого канала сразу же отмечаются признаком DE = 1, но отправляются в СПД, а при превышении порога Be они отбрасываются. В этом случае контрольный интервал времени T = Be/R, где R — скорость доступа к каналу.
В технологии FR определен еще и дополнительный (необязательный) механизм управления потоками кадров, основанный на использовании битов FECN и BECN в кадре FR. Это механизм оповещения конечных пользователей о том, что в коммутаторах СПД возникли перегрузки (переполнение необработанными кадрами).

Бит FECN (Forward Explicit Congestion Bit) кадра извещает принимающую сторону о переполнении в СПД. На основании значения этого бита принимающая сторона должна с помощью протоколов более высоких уровней (TCP/IP, SPX и т. п.) известить передающую сторону о том, что она должна снизить интенсивность отправки пакетов в сеть.
Бит BECN (Backward Explicit Congestion Bit) кадра извещает о переполнении в СПД передающую сторону и является рекомендацией немедленно снизить темп передачи. Обычно он отрабатывается на уровне устройств доступа к СПД FR. Протокол LAP-F не требует от устройств, получивших кадры с установленными битами FECN и BECN, немедленного прекращения передачи кадров в данном направлении. Эти биты служат лишь указанием для протоколов более высоких уровней (TCP, SPX, NCP и т. п.) о необходимости снижения темпа передачи пакетов. Так как управление потоком в разных протоколах организовано по-разному и принимающей, и передающей сторонами, то разработчики протоколов FR учли оба направления, снабдив их предупреждающей информацией о переполнении СПД.
В общем случае биты FECN и BECN могут игнорироваться, но обычно устройства доступа к СПД FR (Frame Relay Access Device — FRAD) отрабатывают по крайней мере признак BECN. При создании коммутируемого виртуального канала параметры качества обслуживания передаются в СПД с помощью протокола LAP-D, который устанавливает виртуальное соединение посредством нескольких служебных пакетов.

Управление доступом

Абонент СПД FR, который хочет установить коммутируемое виртуальное соединение с другим абонентом СПД FR, должен передать через адаптер FRAD по каналу D сообщение SETUP, определяемое следующими параметрами:

- D L C I ;
- адрес назначения;
- максимальный размер кадра в данном виртуальном соединении;
- запрашиваемое значение CIR для двух направлений;
- запрашиваемое значение Bc для двух направлений;
- запрашиваемое значение Be для двух направлений.

Коммутатор, с которым соединен пользователь, сразу же передает ему пакет CALL PROCEEDING — обработка вызова. Затем он анализирует параметры, указанные в пакете, и если коммутатор может их удовлетворить (располагая, естественно, информацией о том, какие виртуальные каналы на каждом порту он уже поддерживает), то пересылает сообщение SETUP следующему коммутатору, который выбирается по таблице маршрутизации.
Протокол автоматического составления таблиц маршрутизации для технологии FR не определен, следовательно, может использоваться фирменный протокол производителя оборудования или же ручное составление таблицы. Если все коммутаторы на пути к конечному узлу согласны принять запрос, то пакет SETUP передается в конечном счете вызываемому абоненту. Вызываемый абонент немедленно передает в сеть пакет CALL PROCEEDING и начинает обрабатывать запрос. Если запрос принимается, то вызываемый абонент передает в сеть новый пакет — CONNECT, который проходит в обратном порядке по виртуальному пути. Все коммутаторы должны отметить, что данный виртуальный канал принят вызываемым абонентом. При поступлении сообщения CONNECT вызывающему абоненту последний должен передать в сеть пакет CONNECT ACKNOWLEDGE. СПД также должна передать вызываемому абоненту пакет CONNECT ACKNOWLEDGE. На этом соединение считается установленным, и по виртуальному каналу могут передаваться данные.

Билет № 37.
Проблемы передачи данных на канальном уровне (Сервис, предоставляемый сетевому уровню, Разбиение на кадры, Контроль ошибок, Управление потоком). Примеры протоколов канального уровня в Internet (протоколы SLIP, PPP, Уровень канала данных в ATM).

Протокол SLIP

SLIP (Serial Line Internet Protocol) — устаревший сетевой протокол канального уровня эталонной сетевой модели OSI для доступа к сетям стека TCP/IP через низкоскоростные линии связи путём простой инкапсуляции IP-пакетов. Используются коммутируемые соединения через последовательные порты для соединений клиент-сервер типа точка-точка.

Принципы работы

- Для установления связи необходимо заранее задать IP-адреса, так как в протоколе SLIP нет системы обмена адресной информацией.
- В принимаемом потоке бит SLIP позволяет определить признаки начала и конца пакета IP. По этим признакам SLIP собирает полноценные пакеты IP и передаёт верхнему уровню. При отправлении IP-пакетов происходит обратная операция — они переформатируются и посимвольно отправляются получателю через последовательную линию.
- Для передачи необходимо использовать конкретную конфигурацию UART: 8 бит данных (8 data bits), без паритета (no parity), аппаратное управление каналом передачи (EIA hardware flow control) или трёхпроводный нуль-модемный кабель (3-wire null-modem — CLOCAL mode).

Недостатки

- Нет возможности обмениваться адресной информацией — необходимость предустановки IP-адресов.
- Отсутствие индикации типа инкапсулируемого протокола — возможно использование только IP.
- Не предусмотрена коррекция ошибок — необходимо выполнять на верхних уровнях, рекомендуется использовать протокол TCP.
- Высокая избыточность — из-за использования стартовых и стоповых битов при асинхронной передаче (+20 %), передачи в каждом SLIP-кадре полного IP-заголовка (+20 байт) и полных заголовков верхних уровней.

Протокол типа точка —точка — PPP
Протокол типа точка—точка — PPP (Point-to-Point Protocol) был разработан по решению комитета IETF (Internet Engineering Task Force) для замены протокола SLIP, обладавшего многочисленными недостатками. Протокол PPP, описанный в RFC 1661, 1662 и 1663, обеспечивает обнаружение ошибок, работает с пакетами разных протоколов сетевого уровня, позволяет динамически выделять IP-адрес только на период соединения, выполняет аутентификацию абонентов и имеет ряд других преимуществ перед протоколом SLIP (Serial Line IP).
Протокол PPP обеспечивает выполнение трех основных групп функций:

1. Распознавание кадров, т.е. однозначное определение конца кадра и начала нового. Здесь же происходит обнаружение ошибок.
2. Управление линией, т.е. активизацию линии, ее проверку, определение основных параметров передачи, корректное завершение передачи со сбросом параметров. Эту группу функций реализует протокол LCP (Link Control Protocol).
3. Определение основных параметров соединения между сетевыми уровнями обменивающихся сторон, что обеспечит независимость от реализации сетевого уровня. Протокол PPP предполагает наличие разных специальных протоколов NCP (Network Control Protocol) на каждом поддерживаемом сетевом уровне.

PPP-кадры имеют формат, очень близкий к формату HDLC-кадров. Основное их различие состоит в том, что PPP —байт-ориентированный, а HDLC —бит-ориентированный. Для HDLC возможен кадр размером в 30,25 байт, а для PPP —нет.

Все PPP-кадры начинаются со стандартного байта 01111110. Поле Address по умолчанию равно 11111111. Поле Control по умолчанию равно 00000011, что означает Unnumbered-кадр, т. е. нумерация передаваемых кадров и подтверждений в их получении здесь не предполагается.
В случае ненадежной среды передачи данных имеется вариант надежной передачи, описанный в RFC 1663.

Так как значения полей Address и Control — константы, то LCP-протокол опускает их, экономя два байта на передаче. В поле Protocol указывается, какой тип пакетов будет в поле Payload. Там допускаются пакеты протоколов LCP, NCP, IP, IPX, Apple Talk и др. Поле Payload, предназначенное для передаваемых данных, имеет переменную длину, которая по умолчанию равна 1 600 байт.

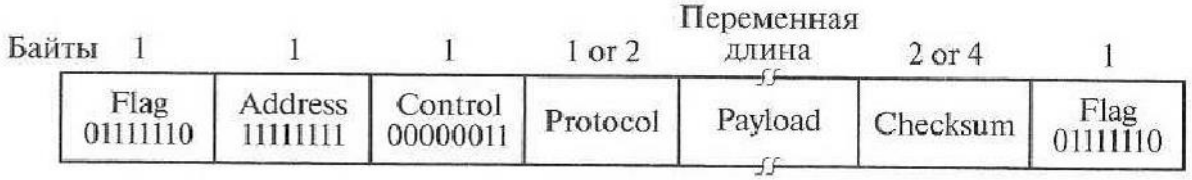


Рис. 4.7. Формат PPP-кадра

Уровень канала данных в СПД ATM

Рассмотрим эталонную модель СПД ATM, которая состоит из трех уровней: физического, ATM-уровня и уровня адаптации. Поверх СПД ATM пользователь может поместить, например стек TCP/IP.
Физический уровень в ATM определяет правила передачи и приема данных в форме потока битов и преобразования их в ячейки. Носителями этого потока могут быть разные физические среды, у ATM здесь нет ограничений.
ATM-уровень отвечает за транспорт ячеек. Он определяет формат ячейки, заголовок, его содержимое, отвечает за установление и поддержание виртуальных соединений. Управление потоком и перегрузками также расположено здесь.
Уровень адаптации (Access Adaptation Layer — AAL) обеспечивает приложениям-пользователям возможность работы в терминах пакетов или подобных им единиц, а не ячеек. Физический уровень в ATM охватывает физический уровень и уровень канала данных в OSI. Поскольку физический уровень ATM на подуровне физической зависимости не предъявляет каких-то специальных требований к физической среде, то сосредоточим внимание на TC-подуровне, т.е. на подуровне подготовки ячеек. Когда прикладная программа посылает сообщение, оно движется вниз по ATM-стеку, получая заголовки, концевики, разбиваясь на ячейки и т.д. Проследим, что происходит с сообщением, когда ячейки достигают TC-подуровня и движутся далее в виде последовательности битов.

Передача ячеек. Первый шаг — вычисление контрольной суммы заголовка. Заголовок состоит из 5 байт, четыре из которых идентифицируют виртуальное соединение и несут контрольную информацию, а один — содержит контрольную сумму. Контрольная сумма защищает только первые четыре байта и не затрагивает данные в ячейке. Контрольная сумма вычисляется как CRC-код, т.е. как остаток от деления содержимого четырех байтов на полином $x^8 + x^2 + x + 1$. К этому остатку добавляется константа 01010101 для повышения надежности в случае, если заголовок содержит много нулей.
Решение защищать контрольной суммой только управляющую информацию было принято с целью сократить затраты на обработку данных на нижних уровнях. Защита собственно данных возлагается на верхние уровни, если это необходимо.
Поскольку контрольная сумма защищает только заголовок, то этот байт так и называется HEC (Header Error Control — контроль ошибки в заголовке). Важным фактором, повлиявшим на выбор такой контрольной схемы, является то, что основной средой в ATM является оптоволокно. Исследования, выполненные компанией AT&T, показали, что оптоволокно — высоконадежная среда, и единичные ошибки происходят в ней с вероятностью менее 1 % . При этом схема HEC прекрасно справляется как с однобитными ошибками, так и множественными. Для надежной передачи ячеек была предложена схема, в которой две последовательные ячейки объединяются через EXCLUSIVE OR, после чего получается новая ячейка, добавляемая в последовательность после первых двух. В результате ячейку, принятую с ошибкой или потерянную, легко можно восстановить. После того как HEC вычислен и добавлен в заголовок, ячейка готова к передаче. При этом среда передачи может быть как синхронной, так и асинхронной. В асинхронной среде ячейка посылается сразу, как только она готова к передаче. В синхронной среде ячейка передается в

соответствии с временными соглашениями. Если ячейки для передачи нет, то TC-подуровень должен сгенерировать специальную ячейку ожидания.
Еще один вид служебных ячеек на TC-подуровне — OAM (Operation And Maintenance). Эти ячейки используются ATM-переключателями для проверки работоспособности системы. Ячейки ожидания обрабатываются TC-подуровнем, а OAM-ячейки передаются на ATM-уровень.
Важной функцией TC-подуровня является также генерирование ячеек в формате физической среды передачи. Это означает, что TC- подуровень генерирует обычную ATM-ячейку и упаковывает ее в кадр надлежащей среды передачи.

Рис. 5.26. Схема состояний автомата для обнаружения заголовка HEC

Прием ячеек. Итак, на выходе TC-подуровень формирует HEC-заголовок, преобразует ячейку в кадр, формирует ATM-ячейки и передает поток битов на физический уровень. На противоположном конце соединение TC-подуровень производит те же самые действия, но в обратном порядке: разбивает поток битов на кадры, выделяет ячейки, проверяет HEC-заголовки и передает ячейки на ATM- уровень.
Самое трудное — выделить кадр из потока битов. На TC-подуровне имеется сдвиговый регистр на 40 бит. Если из этих 40 бит правые 8 бит представляют собой HEC, то последующие 32 бит — это заголовок ячейки. Если это условие не выполнено, то все сдвигается на один бит и проверка повторяется. Этот процесс продолжается до тех пор, пока не будет обнаружен HEC. Представленная схема распознавания заголовка кадра ненадежна. Вероятность того что случайный байт будет выглядеть, как HEC, в этом случае равна 1/256. Для исправления схемы используется автомат. Возможно три состояния автомата: HUNT, PRESYNC, SYNC. В состоянии HUNT автомат ищет HEC. Как только похожий байт найден, автомат переходит в состояние PRESYNC и отсчитывает следующие 53 байт. Если предположить, что найденный HEC — это начало ячейки, то сдвиг на 53 байт приведет к следующему HEC. Проверив последовательно 6 ячеек, автомат переходит в состояние SYNC. Если в состоянии автомата оказалось а плохих последовательных ячеек, происходит переход в состояние HUNT.

Билет № 38.
Протоколы множественного доступа к каналу (динамическое vs статическое выделение канала). Модель системы ALOHA. Сравнение производительности систем: чистая ALOHA, слотированная ALOHA. Протоколы множественного доступа с обнаружением несущей (настойчивые и не настойчивые CSMA, CSMA с обнаружением коллизий).

Статические методы доступа к каналу
При рассмотрении методов множественного доступа к каналу, естественно, возникает идея разделить весь канал на несколько подканалов и каждому потенциальному абоненту предоставить свой подканал - мультиплексирование, или уплотнением канала. Имеется два основных подхода к мультиплексированию: использование частотного (FDM) и временного (TDM) разделения канала. При частотном разделении весь диапазон частот полосы пропускания канала разбивается на поддиапазоны, которые называются подканалами. По каждому подканалу выполняется передача независимо от того, что происходит в других каналах. При временном разделении используется вся полоса пропускания канала для каждого абонента, но при этом время передачи делится на слоты по числу потенциальных абонентов, и каждому из них выделяется свой интервал времени (слот) для передачи. Частотное разделение хорошо работает в условиях, когда число абонентов небольшое и фиксированное и каждый из них обеспечивает плотную загрузку канала. При этом каждому абоненту выделяется своя полоса частот, которую он использует независимо от других. Статическое разделение канала на подканалы является неэффективным решением проблемы доступа при предположении о постоянстве числа абонентов в среднем и нерегулярном трафике у абонентов.

Базовая модель динамического предоставления доступа к каналу
Пять основных предположений, составляющих основу моделей сетей ЭВМ, в которых в качестве СПД используется канал с множественным доступом:
1. Станции. Модель состоит из N независимых станций (компьютеров, телефонов, факс-машин и т. п.). На каждой станции работает пользователь или программа, генерирующие кадры для передачи. Предполагаем, что если кадр сгенерирован, то станция блокируется, и новый кадр не появится, пока не будет передан первый кадр. Это означает, что станции независимы, и на каждой из них работает только одна программа или один пользователь, генерирующие нагрузку с постоянной скоростью.
2. Единственность канала. Канал один и он доступен всем станциям. Все станции равноправны. Они получают кадры и передают кадры только через этот единственный канал. Аппаратные средства всех станций для доступа к каналу одинаковые, но программно можно устанавливать станциям приоритеты.
3. Коллизии. Если во время передачи кадра одной станцией другая станция начала передачу своего кадра, то такой случай будем называть коллизией. Предполагаем, что любая станция может обнаружить коллизию и что кадры, разрушенные при коллизии, должны быть посланы повторно позднее. Кроме коллизий других ошибок передачи нет.
4. Время. Возможны две модели времени — непрерывная и дискретная:
   - непрерывное время. Передача кадра может начаться в любой момент. В сети нет единых часов, которые разбивают время на слоты. Другими словами, время является непрерывной функцией, отображающей интересующие нас действия в сети на множество вещественных чисел;
   - дискретное время. Все время работы канала разбивается на одинаковые интервалы, называемые слотами, В слоте может оказаться нуль кадров, если это слот ожидания, один кадр, если в этом слоте передача кадра прошла успешно, и несколько кадров, если в этом слоте произошла коллизия.
5. Доступ к каналу. Возможны два способа доступа станции к каналу:
   - с обнаружением несущей. Станция прежде чем использовать канал всегда определяет, занят он или нет с помощью несущей — сигнала определенной формы. Когда канал не занят, по нему все время передается такой сигнал, а если канал занят, то сигнал в нем отличается от несущей, и станция не начинает передачу;
   - при отсутствии несущей. Станция ничего не знает о состоянии канала, пока не начнёт использовать его. Она сразу начинает передачу и лишь в ходе передачи обнаруживает коллизию, т.к. сигнал, который она «увидит» в канале, будет отличаться от того сигнала, который станция передала в канал.

Говоря о ***динамическом доступе***, подразумевают, что отсутствует какая-либо фиксированная политика предоставления доступа к каналу для передачи в отличие от статических методов доступа. При этом любая станция может запросить доступ к каналу в любой момент времени, а методы, доступа лишь определяют правила удовлетворения этих запросов.

Методы множественного доступа ALOHA

Система состояла из наземных радиостанций, работающих на одной частоте и связывающих острова между собой. Идея ее конструкции заключалась в том, чтобы позволить в вещательной среде любому количеству пользователей неконтролируемо использовать один и тот же канал.

Чистая ALOHA: любой пользователь, желающий передать сообщение, сразу пытается это сделать. Благодаря тому, что в вещательной среде у него всегда есть обратная связь, т.е. он может определить, пытался ли кто-то еще передавать сообщение на его частоте, отправитель может установить возникновение конфликта при передаче.
Обратная связь в среде ЛВС происходит практически мгновенно. Отправитель при этом должен слушать среду передачи до тех пор, пока последний бит его сообщения не достигнет самого отдаленного получателя. Обнаружив конфликт, отправитель ожидает некоторый случайный отрезок времени, после чего повторяет попытку передачи. Интервал времени на ожидание должен быть случайным, иначе конкуренты, повторяя попытки передачи вызовут коллизию снова. Системы, в которых пользователи конкурируют за получение доступа к общему каналу, называются *системами с состязаниями*.
Неважно, когда произошел конфликт, оба кадра считаются испорченными и должны быть переданы повторно. Контрольная сумма, защищающая данные в кадре, не позволяет различать разные случаи наложения кадров.
Назовем *временем кадра* время, необходимое на передачу кадра стандартной фиксированной длины. Обозначим это время т. Предположим, что число пользователей неограниченно и все они порождают кадры по закону Пуассона со средним числом N кадров за т. Это означает, что вероятность события, при котором будет порождено n кадров за время т, можно записать в виде

$$P[k] = \frac{G^k e^{-G}}{k!},$$

$$P[n] = \frac{\lambda^N e^{-\lambda}}{n!}, \lambda = N.$$

Поскольку при N > 1 очередь на передачу будет только расти и все кадры будут страдать от коллизий, предположим, что 0 < N < 1. Также предположим, что вероятность за время кадра сделать к попыток передачи, как новых, так и ранее не переданных из-за коллизий кадров, распределяется по закону Пуассона со средним значением G. Понятно, что при этом должно выполняться соотношение G >= N, иначе очередь будет расти бесконечно. При слабой загрузке (N ~ 0) будет мало передач, а, следовательно, и коллизий, поэтому G~ N. При высокой загрузке должно выполняться соотношение G> N. При этом пропускная способность канала (S) будет равна числу кадров, которые надо передать, умноженному на вероятность успешной передачи. Если обозначить P_0 вероятность отсутствия коллизий при передаче кадра, то можно записать $S = GP_0$. Рассмотрим внимательно, сколько времени требуется отправителю, чтобы обнаружить коллизию. Пусть он начал передачу в момент времени t_0 и пусть требуется время t, чтобы кадр достиг самой отдаленной станции. Тогда, если в тот момент, когда кадр почти достиг этой отдаленной станции, она начнет передачу (ведь в системе ALOHA. станция сначала передает, а потом слушает), отправитель узнает об этом только через время, равное $t_0 + 2t$. Вероятность появления к кадров при передаче кадра с распределением Пуассона поэтому вероятность, что появится 0 кадров, равна e^{-G}. За двойное время кадра среднее число кадров равно 2G, откуда $P_0 = e^{-2G}$, а так как $S = GP_0$ то пропускная способность канала $S = Ge^{-2G}$.
Максимальная пропускная способность достигается при G = 0,5 при S = 1/2e, что составляет примерно 18 % номинальной пропускной способности системы.

Слотированная ALOHA. Модификация чистой ALOHA, в которой все время работы канала разделяется на слоты. Размер слота при этом должен быть равен максимальному времени кадра. Ясно, что такая организация работы канала требует синхронизации. Кто-то, например одна из станций, испускает сигнал начала очередного слота. Поскольку передачу теперь можно начинать не в любой момент, а только по специальному сигналу, то время на обнаружение коллизии сокращается вдвое. Откуда $S = Ge^{-G}$.
Максимум пропускной способности слотированной ALOHA наступает при G = 1, где S = 1/e, т.е. составляет около 37 % , что вдвое больше, чем у чистой ALOHA.
Рассмотрим, как G влияет на пропускную способность системы. Для этого подсчитаем вероятность успешной передачи кадра за к попыток. Так как e^{-G} — это вероятность отсутствия коллизии при передаче, то вероятность того, что кадр будет передан ровно за к попыток, можно записать в виде

$$P_k = e^{-G}\left(1 - e^{-G}\right)^{k-1}$$

Среднее ожидаемое число повторных передач

$$E = \sum_{k=1}^{\infty} kP_k = \sum_{k=1}^{\infty} ke^{-G}\left(1-e^{-G}\right)^{k-1} = e^{G}$$

Эта экспоненциальная зависимость показывает, что с ростом G резко возрастает число повторных попыток, поэтому незначительное увеличение загрузки канала ведет к резкому падению его пропускной способности.

<u>Протоколы множественного доступа с обнаружением несущей</u>

Протоколы, реализующие идею начала передачи только после определения, занят канал или нет, называются протоколами с обнаружением несущей — CSMA (Carrier Sensitive Multiple Access).

Настойчивые и ненастойчивые CSMA-протоколы

Если канал занят, то станция ждет, а как только он освободился, пытается сразу начать передачу. Если при этом произошла коллизия, станция ожидает случайный промежуток времени и все начинает сначала. Этот протокол называется настойчивым CSMA-протоколом первого уровня, или 1-настойчивым CSMA-протоколом, поскольку станция, следуя этому протоколу, начинает передачу с вероятностью 1, как только обнаруживает, что канал свободен. Здесь важную роль играет задержка распространения сигнала в канале. Всегда существует вероятность того, что, как только одна станция начала передачу, другая станция также стала готова передавать. Если вторая станция проверит состояние канала прежде чем до нее дойдет сигнал от первой станции о том, что она заняла канал, то вторая станция сочтет канал свободным и начнет передачу. В результате возникает коллизия. Чем больше время задержки сигнала, тем больше вероятность такого случая и тем хуже производительность канала.

Однако даже если время задержки сигнала будет равно нулю коллизии все равно могут возникать. Например, если во время передачи готовыми к передаче оказались две станции. В этом случае они подождут, пока ранее начатая передача будет закончена, а затем будут состязаться между собой. Тем не менее этот протокол более эффективен, чем любая из систем ALOHA, так как станция учитывает состояние канала, прежде чем начать действовать.

Другим вариантом CSMA-протокола является ненастойчивый CSMA-протокол. Основное отличие его от предыдущего состоит в том, что готовая к передаче станция опрашивает канал. Если канал свободен, то она начинает передачу. Если же канал занят, то она не будет настойчиво его опрашивать в ожидании, когда он освободится, а будет делать это через случайные отрезки времени. Это несколько увеличивает задержку при передаче сигнала для станции, но общая эффективность протокола возрастает. И, наконец, настойчивый CSMA-протокол уровня p. Который применяется квотированным каналам. Когда станция готова к передаче, она опрашивает канал. Если канал он свободен, то она с вероятностью p передает свой кадр и с вероятностью q = 1 - p ждет следующего слота. Так станция действует, пока не передаст кадр. Если во время передачи происходит коллизия, станция ожидает случайный промежуток времени и опрашивает канал снова. Если при опросе он опять оказывается занятым, станция ждет начала следующего слота, и весь алгоритм повторяется.

CSMA-протокол с обнаружением коллизий

Протоколы этого класса широко используются в локальных сетях. Модель их работы следующая. В момент времени t_0 одна станция заканчивает передачу очередного кадра, а все другие станции, у которых имеется кадр для передачи, начинают передачу. Естественно, в этом случае происходят коллизии, которые быстро обнаруживаются посредством сравнения отправленного сигнала с тем сигналом, который есть на линии. Обнаружив коллизию, станция сразу прекращает передачу на случайный промежуток времени, после чего все начинается сначала. Таким образом, в работе протокола CSMA/CD можно выделить три стадии: состязания, передачи и ожидания (когда нет кадров для передачи).

Рассмотрим подробнее алгоритм состязаний. Определим, сколько времени станции, начавшей передачу, требуется, чтобы обнаружить коллизию. Обозначим т время распространения сигнала до самой удаленной станции на линии. Для коаксиального кабеля длиной в 1 км т = 5 мкс. В этом случае минимальное время для определения коллизии будет равно 2т. Следовательно, станция не может быть уверена, что она захватила канал до тех пор, пока не убедится, что в течение 2т секунд не было коллизий. Поэтому мы будем рассматривать период состязаний как слотированную систему ALOHA со слотом 2т секунд на один бит. Захватив канал, станция может далее передавать кадр с любой скоростью.

обнаружение коллизий — это аналоговый процесс, поэтому, чтобы обнаруживать их, необходимо использовать специальные кодировки на физическом уровне.

Билет № 39.
Протоколы множественного доступа к каналу: Бесконфликтные протоколы (Bit-Map протокол, Адресный счетчик). Протоколы с ограниченным числом конфликтов. Протоколы с множественным доступом и разделением частот.

Бесконфликтные протоколы

при больших т и коротких кадрах коллизии съедают часть пропускной способности канала.
Предположим, что имеется N станций с адресами от 0 до N - 1. Все адреса уникальны. Требуется определить, кто будет владеть каналом, когда закончится текущая передача.

Протоколы с резервированием (bit-map protocol)

В работе протокола выделяется специальный период состязаний, в котором число слотов равно числу станций, подключенных к каналу. Каждая станция, у которой есть кадр для передачи, проставляет 1 в своем слоте. Поскольку рассматривается канал с множественным доступом (т. е. когда все видят, что проходит в канале), то в конце состязаний все станции знают, кто будет передавать кадры и в каком порядке. Передача происходит в том порядке, в каком пронумерованы слоты. При этом, поскольку станции знают, кто будет передавать и в каком порядке, конфликтов не будет. Если станция опоздала с заявкой на передачу, она должна ждать следующего периода состязаний, который начнется по окончании передач, заявленных на предыдущем периоде состязаний. Протоколы, в которых заявки на передачу откладываются и могут быть сделаны лишь в определенные периоды времени, называются протоколами с резервированием.
Теперь рассмотрим производительность этого метода. Пусть N — число станций. Для удобства будем измерять время в количестве слотов состязаний, а также предположим, что передача одного кадра будет занимать ровно d таких слотов, а вероятность готовности станции к передаче распределена равномерно для всех станций. Тогда для станции с небольшим номером, например 0 или 1, время ожидания на передачу в среднем будет равняться 1,5N слотов состязаний, так как она, пропустив начало состязаний, будет ждать 0,5N единиц времени в первом периоде состязаний и единиц времени — во втором периоде. Станции со старшими номерами будут ожидать в среднем N/2 слотов до начала передачи. Таким образом, в среднем любая станция должна будет ждать N слотов состязаний до передачи. При небольшой нагрузке накладные расходы на передачу одного кадра составят N бит, а эффективность передачи одного кадра—d/(d + N), где N — накладные расходы на передачу кадра. При плотной загрузке, когда практически каждая станция каждый раз что-то посылает, накладные расходы будут составлять 1 бит на кадр, т.е. d/(d +1). Средняя задержка кадра будет равна средней задержке кадра внутри очереди в станции плюс N(d + 1)/2 слотов ожидания, когда кадр достигнет заголовка очереди к каналу. Отсюда видно, что с ростом N, хотя накладные расходы на передачу одного кадра падают, задержка кадра в канале существенно возрастает и эффективность падает.

Протокол двоичного адреса

Один из недостатков протоколов с резервированием — это затраты на слоты состязаний, составляющие 1бит на станцию. При коротких кадрах это накладно. Повысить эффективность применения канала можно, используя двоичное представление адреса станции. В этом методе каждая станция, готовая к передаче, на стадии состязаний выставляет свой адрес бит за битом, начиная со старшего разряда. Эти разряды подвергаются логическому сложению. Если станция выставила на очередном шаге 0, а результат логического сложения 1, то она должна ждать, т.е. в текущих состязаниях она далее участия не принимает.
Эффективность использования канала в этом методе составляет d/(d + lnN). Если структура заголовка кадра при этом выбирается таким образом, чтобы его можно было использовать для выбора очередной станции для передачи, то эффективность использования канала достигнет 100% , поскольку слагаемое InN уходит.
Этот метод, как и предыдущий, имеет один существенный недостаток — он несправедливый: чем больше номер станции, тем скорее она захватит канал.

Протоколы с ограниченным числом конфликтов

Была попытка создать протокол, обеспечивающий возможность использования состязаний при небольших нагрузках и бесконфликтных методов — при высоких. Такие протоколы были созданы, и называются они протоколами с ограниченным числом конфликтов.

Симметричные протоколы с ограниченным числом состязаний

Пусть имеется к станций, каждая из которых с вероятностью p готова передать кадр. В этом случае вероятность, что какая-то станция успешно передаст свой кадр, равна $кр(1-р)^{к-1}$ и эта вероятность достигает максимума при p = 1/k. Тогда вероятность передачи сообщения какой-либо станцией равна $((к - 1)/к)^{к-1}$. При небольшом числе станций шансы передать кадр достаточно велики, но с ростом числа станций эти шансы резко падают. Единственным способом увеличить шансы на передачу является сокращение числа конфликтов. Для этого в протоколах с ограниченным числом состязаний все станции разбивают на непересекающиеся группы. Каждой группе присваивается номер. За слот с номером 0 состязаются только станции из группы 0. Если передавать в этой группе нечего или была коллизия, то начинаются состязания за слот 1 между членами группы 1 и т.д. Основную сложность в этом методе представляет распределение станций по группам.

Адаптивный древовидный протокол

Данный протокол, конкретизирующий способ распределения станций по группам, позволяет эффективно находить оптимальное число станций в группе. За слот 0 борются все станции. Если какая-то из них победила — хорошо, а если нет, то за слот 1 борются только станции поддерева с корнем в вершине 2. Если какая-то из них победила, то следующий слот резервируется для станций поддерева с корнем в вершине 3. Если какая-то из них победила, то за следующий слот борются станции поддерева с корнем в вершине 4, и т.д. Когда число станций велико и все они готовы передавать, то вряд ли целесообразно начинать поиск с уровня 0 дерева. Перенумеруем уровни дерева: на уровне 0 — вершина 1, на уровне 1 — вершины 2 и 3 и т.д. Заметим, что на уровне i располагается 2^i вершин, т.е. число станций, являющихся листьями в поддереве с корнем на уровне i равно $N/2^i$. Пусть число станций, готовых к передаче, нормально распределено. Обозначим это число q. Тогда число станций, готовых к передаче и расположенных ниже узла уровня i, будет равно $2^{-i}q$. Необходимо подобрать такое соотношение между i и q, при котором число конкурирующих станций будет равно 1, т.е. будет выполняться условие $2^{-i}q = 1$, или $\log_2 q = i$.

Билет № 40.
Сотовая связь: пейджинг, сотовые и радиотелефоны (система AMPS, GSM, GPRS, UMTS, CDMA).

Первые сотовые телефонные системы были аналоговыми, например система AMPS (Advanced Mobil telephone System). Им на смену пришли цифровые системы, которые составили второе поколение сотовых систем. В настоящее время происходит переход на сотовые системы 3G — системы третьего поколения. Для перехода на цифровые системы было как минимум три причины:

- природа оцифрованных данных неважна, поэтому можно интегрировать в одном и том же канале и голос, и факс, и данные;
- для оцифрованных данных имеются хорошие алгоритмы сжатия, обнаружения и исправления ошибок;
- данные в цифровой форме можно шифровать в целях безопасности.

Европейская система GSM, свободна от каких-либо компромиссов ради достижения совместимости с уже существующими системами.
В Европе для нее используют частоты 900 и 1800 МГц, а в США — 1 900 МГц. Европейцы создали единую цифровую систему, известную как GSM (Global System for Mobile communications) , которая была введена в действие раньше американских и японских аналогов.
Идея организации любой сотовой системы связи, в том числе и сети GSM, очень проста. Вместо того чтобы охватить сразу всю необходимую территорию небольшим числом каналов, эту территорию разбивают на небольшие части — соты. В каждой соте используют свой набор каналов, но таким образом, чтобы частоты каналов соседних сот не пересекались, т.е. чтобы у них не было общих частот. Такая организация системы дает выигрыш в использовании частот вследствие возможности их повторного использования, а значит, увеличивается емкость сети — число одновременно обслуживаемых пользователей. Кроме того, в системе можно использовать маломощные сигналы, а следовательно, передатчик может быть компактным, так как не требуется мощных источников питания. Если в каких-то сотах из-за большого числа пользователей отказы в соединениях становятся слишком частыми, то эти соты можно разделить на несколько новых. Каждая сота имеет базовую станцию (BS) — базу, состоящую из компьютера и приемно-передающей аппаратуры. Несколько BS подключаются к Центру мобильной коммутации (MSC). В небольших системах может быть достаточно одного такого центра, а в больших системах их может потребоваться несколько. MSC-центры соединяются друг с другом и с обычной наземной телефонной сетью и при необходимости коммутируют звонок с мобильного телефона на обычный телефон.
При перемещении телефона ближайшие базовые станции сравнивают уровень сигнала, поступающего от него, и та база, на которой этот уровень выше, чем на других, берет этот сигнал под свой контроль. При этом телефон получает сообщение об изменении базы. Все каналы в соте подразделяются на следующие четыре категории:

- управляющие;
- для сообщений;
- для установки доступа и распределения каналов;
- для данных (голоса, факса и пр.).

Когда телефон включают, он начинает сканировать запрограммированный в нем список каналов управления, чтобы обнаружить наиболее мощный сигнал. По информации из управляющего канала он узнает о распределении каналов для сообщений, установки соединений и доступа, передачи данных. Затем телефон сообщает через базовую станцию центру мобильной коммутации свою уникальную идентифицирующую информацию. Когда базовая станция получает такую информацию от телефона, она запрашивает у своего MSC-центра информацию о новом клиенте и сообщает домашнему MSC, т.е. MSC, к которому приписан этот телефон, о его текущем местоположении. Обычно такая перерегистрация телефона происходит периодически.
Телефон по каналу установки доступа посылает информацию о себе и о телефоне вызываемого абонента. Получив запрос, базовая станция информирует о нем MSC. Если вызываемый абонент является абонентом компании, которой принадлежит MSC, то MSC ищет свободный канал для данных. Если такой канал найден, то MSC информирует о нем вызывающий телефон по каналу управления. Вызывающий телефон переключается на прием по указанному каналу и ждет, когда на вызываемом телефоне поднимут трубку. Входящий звонок обрабатывается несколько иначе. В режиме ожидания телефон постоянно следит за каналом сообщений: т.е. не появится ли там сообщение для него. Когда вызывающий телефон сгенерировал запрос, то от MSC поступает запрос на домашний MSC вызываемого телефона, чтобы определить, в какой

соте находится этот телефон. Пакет с вызовом направляется последней базовой станции, зарегистрировавшей телефон с искомым номером, например 75. Базовая станция распространяет по каналу сообщений специальное сообщение. Вызываемый телефон отвечает по каналу управления специальным пакетом. Тогда базовая станция шлет по каналу управления пакет, после чего вызываемый телефон переключается на канал 8 и начинает звонить до тех пор, пока на нем не нажмут кнопку Прием.

Теперь рассмотрим эту схему применительно к стандарту GSM. Итак, GSM —это полностью цифровая система. При этом в любой стране может быть одна или несколько функционирующих GSM-сетей, причем каждая такая сеть является региональной мобильной сетью оператора (PLMN). Зона действия каждой PLMN-сети ограничена национальными границами. Впрочем, в одной стране может быть несколько PLMN-сетей. GSM-пользователь заключает контракт с одной из PLMN-сетей, называемой домашней, в котором указаны услуги, доступные этому пользователю. При желании пользователь во время работы может выбрать другую PLMN-сеть, если ему доступны ее услуги. Терминал пользователя, который в GSM называется мобильной станцией —MS, а в просторечье —трубкой, обеспечивает пользователю такой выбор и показывает список доступных PLMN-сетей. Выбор из этого списка пользователь может сделать сам явно или MS-терминал сделает это автоматически с помощью заложенного в него программного обеспечения. Как и в AMPS-системе, в GSM территория разбивается на области, обслуживаемые центром мобильной коммутации (MSC). Оператор PLMN-сети абсолютно свободен в разбиении области действия MSC-станции на соты. При этом у каждой PLMN-сети есть логически единая база данных, называемая Home Location Registers (HLR), где хранится информация обо всех пользователях, для которых эта PLMN-сеть является домашней. Физически HLR-база может быть распределенной. У каждой MSC-станции имеется база данных визитеров (Visitor Location Registers —VLR). Обычно одна VLR-база обслуживает одну MSC-станцию, но она может обслуживать и несколько станций. Базы данных HLR и VLR обеспечивают отслеживание текущего местонахождения каждого MS-терминала, находящегося в зоне действия MSC-станции, запрашиваемых услуг и т.д.

Мобильная станция GSM подразделяется на две части, одна из которых обеспечивает радиоинтерфейс, а другая —интерфейс с базами HLR и VLR и содержит информацию, идентифицирующую пользователя (Subscriber Identify Module —SIM). Идентифицирует пользователя SI M-карта, а не MS-терминал, поэтому ее можно вынуть из одного MS-терминала и вставить в другой. Каждая SIM-карта уникальна в системе GSM и связана с IMSI-идентификатором (International Mobil System Identify). На SIM-карте хранятся идентификационная информация, список услуг, список выбираемых PLMN-сетей и т.п. Она также защищена паролем (PIN —Personal Identification Number). Вставив свою SIM-карту в трубку, пользователь тем самым персонифицирует ее. Благодаря SIM-карте поддерживается роуминг. т.е. доступ к услугам связи в чужой PLMN-сети.

Теперь рассмотрим, как в GSM отслеживаются перемещения пользователей. Когда MS-терминал входит в новую область регистрации, информация о нем заносится в VLR-базу, и он получает TMSI-идентификатор (Temporary Mobil Subscriber Identify). TMSI-идентификатор короче IMSI-идентификатора, поэтому именно он передается при взаимодействии MS-терминала и VLR-базы. TMSI-идентификатор действует только в зоне MSC-станции, ассоциированной с VLR-базой, выдавшей его. IMSI и TMSI —это внутренние идентификаторы системы, связанные с SIM-картой. Для соединения с абонентом используется телефонный номер, который в GSM называется Mobil Subscriber Integrated Service Digital Network Number (MSISDNM). MS-терминал всегда может определить, находится он в старой или В новой области регистрации. Это происходит благодаря периодически рассылаемой BS-станцией информации внутри обслуживаемой ею соты. Если MS-терминал обнаруживает, что оказался в новой области, то он инициирует запрос на обновление регистрации, в котором сообщает идентификатор предыдущей области и TMSI-идентификатор, полученный им в этой области. Этот запрос BS-станция передает в MSC-центр, который, в свою очередь, передает его в VLR-базу. Эта новая VLR-база инициирует запрос к старой VLR-базе с просьбой предоставить IMSI-идентификатор терминала, соответствующий указанному TMSI-идентификатору. Получив от старой VLR-базы необходимую информацию, новая VLR-база начинает процедуру идентификации MS-терминала по полученной информации. Если процедура идентификации прошла успешно, то новая VLR-база, используя IMSI-идентификатор терминала, определяет адрес его HLR-базы. Эта процедура весьма близка к аналогичной процедуре в AMPS-системе и основное ее отличие от AMPS-аналога состоит в повышенной информационной безопасности. Так, например, идентификация пользователя и доступных ему услуг происходит здесь на основе информации, получаемой новой VLR-базой как от старой VLR-базы, так от HLR-базы идентифицируемого MS-терминала, а не только от HLR-базы, как в AMPS-системе. Процедура установления соединения в GSM-системе аналогична процедуре установления соединения в AMPS-системе, Стандарт GSM занимает более 5000 страниц, и здесь мы приводим лишь самое общее его описание. В большинстве стран для GSM используются частоты 900 и 1 800 МГц, а в США из-за особенностей национального распределения частот применяется другой диапазон. При этом в каждой соте выделается до 200 каналов: 124 канала —для

абонентов, а остальные —резервные и служебные. Каждый канал состоит из двух полос: входящей —от базы к мобильной станции (терминалу) и исходящей —от мобильной станции (терминала) к базе. Каждая полоса имеет ширину в 200 кГц (рис. 5.27). Каждый из 124 частотных каналов может поддерживать до восьми соединений, используя технику TDM-мультиплексирования. Теоретически может поддерживаться до 992 соединений одновременно.
Как уже говорилось, каждый канал поддерживает восемь разных соединений с помощью мультиплексирования с разделением по времени (TDM-метод). Каждому MS-терминалу выделяется одно соединение, т.е. один временной слот на одном из каналов. Однако для обеспечения качества передачи и устранения помех из-за частотных конфликтов между каналами используются не все каналы.

GPRS-служба

Для высокоскоростной передачи данных посредством существующих GSM-сетей была разработана служба пакетной передачи данных по радиоканалу GPRS (General Packet Radio Service). Кроме повышения скорости (максимум составляет 171,2 Кбит/с) новая система предполагает иную схему оплаты услуги передачи данных, т.е. при использовании GPRS-службы расчет за услугу производится пропорционально объему переданной информации, а не по времени использования канала. К тому же GPRS-служба более рационально использует выделяемую полосу частот, т.е. особо не вдаваясь в технические тонкости, можно сказать, что пакеты данных передают одновременно по нескольким соединениям (именно за счет одновременного использования нескольких соединений и получается выигрыш в скорости) в паузах между передачей речи. Поскольку голосовой трафик имеет безусловный приоритет перед трафиком данных, скорость передачи информации определяется не только возможностями сетевого и абонентского оборудования, но и загрузкой сети. Ни один канал GPRS-службы не занимается под передачу данных полностью — и это является основным качественным отличием новой технологии от описанных ранее.

Внутренняя организация GPRS-службы

Если говорить о программном обеспечении, то оно нуждается в замене или обновлении практически полностью, начиная с баз HLR и VLR и заканчивая базовыми станциями (BS). В частности, для временных кадров каналов GSM вводится режим многопользовательского доступа, а в HLR-базе появляется новый параметр — Mobile Station Multislot Capability (число соединений, которое одновременно может занимать мобильный телефон абонента).
Ядро системы GPRS (GPRS Core Network) состоит из двух основных узлов: узла поддержки GPRS-сервиса (узел Serving GPRS Support Node — SGSN) и шлюзового узла GPRS (Gateway GPRS Support Node - GPRS). SGSN контролирует доставку пакетов данных пользователям, взаимодействует с HLR-базой собственных абонентов сети, проверяя разрешены ли запрашиваемые пользователями услуги, ведет мониторинг находящихся в режиме on-line пользователей, организует регистрацию вновь появившихся в зоне действия сети абонентов и т. п. Назначение GGSN-узла видно из его названия — это шлюз между сотовой сетью и внешними информационными магистралями.
Основной задачей GGSN-узла является коммутация данных, идущих через SGSN-узел к абоненту и от абонента. Вторичными функциями GGSN-узла являются адресация данных, динамическая выдача адресов в Интернет (IP-адресов), а также отслеживание информации о внешних сетях и собственных абонентах (в том числе тарификация услуг).
В GPRS-службу заложена хорошая масштабируемость: при появлении новых абонентов оператор может увеличивать число SGSN-узлов, а при эскалации суммарного трафика — добавлять в систему новые GGSN-узлы.
Еще одной составной частью системы GPRS является блок контроля пакетной передачи (Packet Control Unit — PCU). Блок PCU стыкуется с контроллером базовых станций (BSC) и отвечает за направление трафика данных непосредственно от BSC к SGSN.

Качество сервиса в GPRS(Quality of Service — QoS)

В GPRS существует несколько классов QoS, различающихся по следующим признакам:

- по приоритету (высокий, средний и низкий приоритет данных);
- надежности (разделение на три класса по числу возможных ошибок передачи, потерянных пакетов и т.п.);
- задержкам (задержки информации вне GPRS-сети в расчет не принимаются);

- количественным характеристикам (пиковое и среднее значения скорости).

Стандарт услуги GPRS предусматривает два режима соединений:

- PTP (Point-To-Point — точка—точка);
- PTM (Point-To-Multipoint — вещание).

Режим PTM (вещания), в свою очередь, подразделяется на два класса:

- PTM-M (PTM-Multicast) — передача необходимой информации всем пользователям, находящимся в определенной географической зоне;
- PTM-G (PTM-Group Call) — отправка данных определенной группе пользователей.

Метод множественного доступа на основе разделения кодов

Метод множественного доступа на основе разделения кодов — CDMA (Code Division Multiple Access) основан на следующей идее: каждый участник связи может использовать всю полосу пропускания канала в соте за счет применения метода прямого расширения спектра передачи.
В CDMA-системе каждый бит сообщения кодируется последовательностью из m частиц (чипов). Бит со значением 0 передается инвертированной последовательностью частиц, а бит со значением 1 — прямой. Каждой мобильной станции присваивается уникальный код — последовательность частиц для 0 и для 1. Ясно, что такая техника возможна, только если при увеличении объема передаваемой информации будет пропорционально увеличиваться ширина полосы пропускания. Кроме того, поскольку каждая станция имеет уникальную последовательность частиц, не требуется дополнительного шифрования. Идея уникальности последовательности частиц для каждой станции основана на ортогональных кодах. Суть этих кодов состоит в следующем: если рассмотреть последовательности частиц для станции как векторы S и T соответственно, то можно записать

$$(S,T)=\frac{1}{m}\sum_{i=1}^{m}S_iT_i=0.$$

Билет № 41.
Стандарты IEEE 802.x для локальных и муниципальных сетей: Стандарт IEEE 802.3 и Ethernet (кабели, способ физического кодирования, алгоритм вычисления задержки, MAC подуровень, производительность).

Стандарт IEEE 802.3 и Ethernet

Стандарт IEEE 802.3 относится к 1-настойчивым протоколам CSMA/CD для локальных сетей. Прежде чем начать передачу, станция, использующая такой протокол, опрашивает канал. Если он занят, то она ждет и как только он освободится, она начинает передачу. Если несколько станций одновременно начали передачу, то возникает коллизия. Тут же передача прекращается. Станции ожидают некоторый случайный отрезок времени, и все начинается сначала.
Чтобы увеличить длину сегмента, используются репитеры. Это устройство физического уровня, которое отвечает за очистку, усиление и передачу сигнала. Репитеры не могут отстоять более чем на 2,5 км, и на одном сегменте их не может быть более четырех.

При использовании **Манчестерского кода** весь период передачи бита разбивается на два равных интервала. При передаче 1 передается высокий сигнал в первом интервале и низкий - во втором. При передаче 0 - наоборот. Такой подход имеет переход в середине передачи каждого бита, что позволяет синхронизироваться приемнику и передатчику. Недостатком такого подхода является то, что пропускная способность канала падает вдвое по сравнению с прямым кодированием. При использовании **дифференциального манчестерского кода** при передаче 1 в начале передачи нет различия в уровне с предыдущим интервалом передачи, т.е. нет перепада в уровне в начале каждого интервала, а при передаче 0 - есть. Этот способ кодирования обладает лучшей защищенностью, чем просто манчестерский код, но требует более сложного оборудования

IEEE 802.3 предписывает, что кадр не может быть короче 64 байт. Если длина поля данных недостаточна, то поле Pad компенсирует нехватку длины. Ограничение на длину кадра связано со следующей проблемой. Если кадр короткий, то станция может закончить передачу прежде, чем начало кадра достигнет самого отдаленного получателя. В этом случае она может пропустить коллизию и ошибочно считать, что кадр доставлен благополучно.

Двоичный экспоненциальный алгоритм задержки

Теперь рассмотрим, как определяется случайная величина задержки при возникновении коллизий. При возникновении коллизии время разбивается на слоты длиной, соответствующей наибольшему времени распространения сигнала в оба конца (2τ). Для 802.3, как уже было указано, это время при длине линии 2,5 км и четырех репитерах равно 51,2 мксек.

При первой коллизии станции, участвовавшие в ней, случайно выбирают 0 или 1 слот для ожидания. Если они выберут одно и то же число, то коллизия возникнет опять. Тогда выбор будет происходить среди чисел 0, 2i, 1, где i - порядковый номер очередной коллизии.

После 10 коллизий число слотов достигает 1023 и далее не увеличивается, после 16 коллизий Ethernet-контроллер фиксирует ошибку и сообщает о ней более высокому уровню стека протоколов.
Если передача кадра средней длины занимает m сек, то при условии большого числа станций, постоянно имеющих кадры для передачи, эффективность канала равна m/(m+2t/A) A-вероятность, что станция захватит канал, 2т-длительность слота. хотя с ростом длины кадра эффективность канала растет, время задержки кадра в системе также увеличивается.

Название	Тип кабеля	Максимальная длина сегмента	Кол-во узлов на сегмент	Преимущества	Подключение
10Base5	Толстый коаксиал	500 м	100	Подходит для магистралей	Трансивер на кабеле соединяется с компьютером трансиверным кабелем. Его длина не должна превосходить 50 метров. Он состоит из 5 витых пар.
10Base2	Тонкий коаксиал	200 м	30	Самый дешевый	Т-образное соединение BNC коннектором. трансивер расположен на контроллере в компе.
10Base-T	Витая пара	100 м	1024	Простое обслуживание	Подключение витухой к хабу
10Base-F	Оптоволокно	2000 м	1024	Идеально для соединения зданий	

Шина с маркером 802.4

Физически шина с маркером имеет линейную или древовидную топологию. Логически станции объединены в кольцо (рисунок 4-22), где каждая станция знает своего соседа справа и слева. Когда кольцо инициализировано, станция с наибольшим номером может послать первый кадр. После этого она передает разрешение на передачу кадра своему непосредственному соседу, посылая ему специальный управляющий кадр - маркер. Передача кадра разрешена только той станции, которая владеет маркером.

Билет № 42.
Системы FDDI: Основы технологии FDDI. Протоколы технологии FDDI.

Основы технологии FDDI

Разработчики технологии FDDI ставили перед собой в качестве наиболее приоритетных следующие цели:

- повысить скорость передачи данных до 100 Мбит/с;
- повысить отказоустойчивость СПД за счет стандартных процедур восстановления после отказов различного рода — повреждения кабеля, некорректной работы узла, концентратора, возникновения высокого уровня помех на линии и т.п.;
- максимально эффективно использовать потенциальную пропускную способность СПД как для асинхронного, так и для синхронного трафиков.

СПД FDDI строится на основе двух оптоволоконных колец, по одному из которых трафик направлен по часовой стрелке, а по другому — против часовой стрелки. При этом в случае выхода из строя одного из колец его трафик может быть запущен через второе кольцо. Если оба кольца окажутся поврежденными в одном и том же месте, то они могут быть объединены в одно кольцо.
Использование двух колец — это основной способ повышения отказоустойчивости FDDI, и чтобы им воспользоваться, узел должен быть подключен к обоим кольцам. В нормальном режиме работы СПД данные проходят через все узлы и все участки кабеля первичного (Primary) кольца, поэтому этот режим называется Thru — сквозным или транзитным. Вторичное кольцо (Secondary) в этом режиме не используется. При образовании общего кольца из двух колец передатчики станций по-прежнему остаются подключенными к приемникам соседних станций, что позволяет правильно передавать и принимать информацию соседним станциям. В стандартах FDDI отводится много внимания различным процедурам, которые позволяют определить наличие отказа, а затем произвести необходимую реконфигурацию. СПД FDDI может полностью восстанавливать свою работоспособность в случае единичных отказов ее элементов. При множественных отказах СПД распадается на несколько не связанных СПД. Кольца в СПД FDDI рассматриваются как общая разделяемая среда передачи данных, поэтому для нее определен специальный метод доступа. Этот метод называется методом кольца с маркером. Станция может начать передачу своих собственных кадров данных только в том случае, если она получила от предыдущей станции специальный кадр — маркер доступа. После этого она может передавать свои кадры в течение времени, называемого временем удержания маркера (Token Holding Time — THT). После истечения значения THT станция обязана завершить передачу своего очередного кадра и передать маркер доступа следующей станции. Если же в момент получения маркера у станции нет кадров для передачи, то она немедленно передает маркер следующей станции.
У каждой станции имеется предшествующий сосед (upstream neighbor) и последующий сосед (downstream neighbor), определяемые ее физическими связями и направлением передачи информации. Каждая станция постоянно принимает передаваемые ей предшествующим соседом кадры и анализирует их адрес назначения. Если адрес назначения не совпадает с ее собственным, то она транслирует кадр своему последующему соседу. Если же адрес кадра совпадает с адресом станции, то она копирует этот кадр в свой внутренний буфер, проверяет его корректность (с помощью контрольной суммы), передает его поле данных для последующей обработки протоколу, расположенному выше FDDI-уровня (например, IP), а затем передает исходный кадр по кольцу последующей станции.
В передаваемом по кольцу кадре станция-получатель отмечает три признака: распознавания адреса, копирования кадра, и отсутствия или наличия в нем ошибок. Станция-отправитель проверяет эти признаки и таким образом определяет, дошел ли кадр до станции получателя и не был ли он при этом поврежден. Процесс восстановления информационных кадров не входит в функции протоколов FDDI, этим должны заниматься протоколы более высоких уровней.

Протоколы технологии FDDI

В технологии FDDI используются протокол физического уровня и протокол подуровня доступа к среде (MAC) канального уровня. В технологии FDDI используется также протокол 802.2 подуровня управления каналом данных (LLC), определенного в стандартах IEEE 802.2 и ISO 8802.2, при котором узлы работают в дейтаграммном режиме, т.е. без установления соединений и без восстановления потерянных или поврежденных кадров.
Физический уровень подразделяется на независимый от среды подуровень PHY (Physical) и зависящий от среды подуровень PMD (Physical Media Dependent). Работу всех уровней контролирует протокол управления

станцией SMT (Station Management), Напомним, что мы уже сталкивались с таким делением на подуровни при рассмотрении семейства Ethernet IEEE 802.3. Аналогично организован физический уровень в СПД АТМ. Подуровень PMD обеспечивает необходимые средства для передачи данных от одной станции к другой по оптоволокну. В его спецификации определяются:

- требования к мощности оптических сигналов и к многомодовому оптоволоконному кабелю;
- требования к оптическим обходным переключателям (optical bypass switches) и оптическим приемопередатчикам;
- параметры оптических разъемов MIC (Media Interface Connector) и их маркировка;
- длина волны 1 300 нм, на которой работают приемопередатчики;
- представление сигналов в оптических волокнах в соответствии с кодировкой NRZI-1

Подуровень PHY выполняет кодирование и декодирование данных, циркулирующих между подуровнями MAC и PMD, а также обеспечивает тактирование информационных сигналов. В его спецификации определены:

- правила кодирования информации в соответствии со схемой 4B/5B (которая рассматривалась при изучении Fast Ethernet);
- правила тактирования сигналов;
- требования к стабильности тактовой частоты 125 МГц;
- правила преобразования информации из параллельной формы в последовательную.

Подуровень MAC ответственен за управление доступом к сети, а также за прием и обработку кадров данных. В его спецификации определены:

- правила захвата и передачи маркера;
- правила формирования кадра;
- правила генерации и распознавания адресов;
- правила вычисления и проверки 32-разрядной контрольной суммы.

Уровень SMT выполняет все функции по управлению и мониторингу остальных уровней стека протоколов FDDI. В управлении кольцом принимает участие каждый узел сети FDD1, поэтому все узлы обмениваются специальными кадрами SMT для управления сетью. В спецификации SMT определены:

- алгоритмы обнаружения ошибок и восстановления после сбоев;
- правила мониторинга работы кольца и станций;
- алгоритм управления кольцом;
- процедуры инициализации кольца.

Билет № 43.
Системы FDDI: Типы узлов и правила их соединения. Функции MAC-подуровня.

Типы узлов и правила их соединения

Все станции в СПД FDDI подразделяются на конечные станции и концентраторы, а также их различают по способу присоединения к первичному и вторичному кольцам и по числу MAC-адресов у одной станции. Для того чтобы передавать собственные данные в кольцо (а не просто ретранслировать данные соседних станций), станция должна иметь в своем составе хотя бы один MAC-узел со своим уникальным MAC-адресом. Станции могут не иметь ни одного MAC-узла, а значит, участвовать только в ретрансляции чужих кадров, но обычно все станции в СПД FDDI, даже концентраторы, имеют хотя бы один MAC-узел. Концентраторы используют MAC-узел для захвата и генерации служебных кадров.

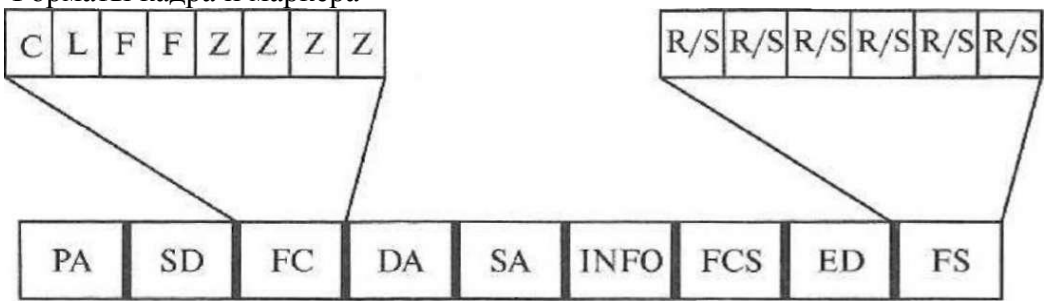
Функции MAC-подуровня

Подуровень MAC выполняет в технологии FDDI следующие функции:

- поддерживает сервисы для подуровня LLC;
- формирует кадр определенного формата;
- управляет процедурой передачи маркера;
- управляет доступом станции к среде;
- обеспечивает адресацию станции;
- копирует кадры, предназначенные для данной станции, в буфер и уведомляет подуровень LLC и блок управления станцией SMT о прибытии кадра;
- генерирует контрольную сумму кадра с помощью CRC-кода и
- проверяет ее у всех кадров, циркулирующих по кольцу;
- удаляет из кольца все кадры, которые сгенерировала данная станция;
- управляет таймерами, которые контролируют логическую работу кольца, т.е. таймером удержания маркера, таймером оборота маркера и т.д.;
- осуществляет мониторинг определенных событий, что помогает обнаружить и локализовать неисправности;
- определяет механизмы, используемые кольцом для реакции на ошибочные ситуации, т.е. на повреждение кадра, потерю кадра, потерю маркера и т.д.

В каждом блоке подуровня MAC параллельно работают два процесса: процесс передачи символов (MAC Transmit) и процесс приема символов (MAC Receive). За счет этого подуровень MAC может одновременно передавать символы одного кадра и принимать символы другого кадра.

Форматы кадра и маркера



В СПД FDDI информация передается в форме двух блоков данных: кадра и маркера. Рассмотрим назначение полей кадра.
Преамбула (PA). Любой кадр должен предваряться преамбулой, состоящей как минимум из 16 символов Idle (1). Эта последовательность сигналов предназначена для синхронизации приемника и передатчика кадра.
Начальный ограничитель (Starting Delimiter — SD). Состоит из пары символов JK, которые позволяют однозначно определить границы для остальных символов кадра.
Поле управления (Frame Control — FC). Идентифицирует тип кадра и детали работы с ним. Имеет 8-битовый формат и передается с помощью двух символов. Состоит из подполей, обозначаемых CLFFZZZZ, которые имеют следующее назначение:

- C — определяет тип трафика, который переносит кадр (синхронный — значение 1, асинхронный — значение 0);
- L — определяет длину адреса кадра, который может занимать 2 или 6 байт;
- FF — определяет тип кадра (01 — обозначение кадра LLC (пользовательских данных), 00 — обозначение служебного кадра MAC-подуровня). Служебными кадрами MAC-подуровня являются кадры процедуры инициализации кольца Claim Frame, кадры процедуры сигнализации о логической неисправности Beacon Frame и кадры процедуры управления кольцом SMT Frame;
- ZZZZ — детализирует тип кадра.

Адрес назначения (Destination Address — DA). Идентифицирует станцию (уникальный адрес) или группу станций (групповой адрес), которым предназначен кадр. Может содержать 2 или 6 байт.
Адрес источника (Source Address, SA). Идентифицирует станцию, сгенерировавшую данный кадр. Это поле должно быть той же длины, что и поле адреса назначения.
Информация (INFO). Содержит информацию, относящуюся к операции, указанной в поле управления. Может иметь длину от 0 до 4 478 байт (от 0 до 8 956 символов). Стандарт FDDI допускает размещение в этом поле информации о маршруте для алгоритма маршрутизации от источника (Source Routing), устанавливаемой стандартом 802.5, который определяет работу СПД Кольцо с маркером (Tocken Ring). При этом в два старших бита поля адреса источника SA помещается комбинация 102 — групповой адрес, не имеющая отношения к адресу источника, а обозначающая присутствие информации о маршруте в поле данных.
Контрольная последовательность (Frame Check Sequence — FCS). Содержит 32-битовую последовательность, вычисленную стандартным методом CRC-32, принятым и для других протоколов IEEE 802. Контрольная последовательность охватывает поля FC, DA, SA, INFO и FCS.
Конечный ограничитель (Ending Delimiter — ED). Содержит единственный символ Terminate (T), обозначающий границу кадра. За ним располагаются признаки статуса кадра.
Статус кадра (Frame Status — FS). Первые три признака в этом поле являются индикаторами ошибки (Error — E), распознавания адреса (Address recognized — A) и копирования кадра (Frame Copied — C). Каждый из этих индикаторов кодируется одним символом, причем нулевое состояние индикатора обозначается символом Reset (R), а единичное — символом Set (S). Стандарт позволяет производителям оборудования добавлять свои индикаторы после трех обязательных.

Операции MAC-подуровня

Захват маркера. Если станция имеет право захватить маркер, то после ретрансляции на выходной порт символов PA и SD маркера она удаляет из кольца символ FC, по которому распознала маркер, а также конечный ограничитель ED. Затем вслед за уже переданным символом SD станция передает символы своего кадра. Таким образом, станция, как и прежде, формирует новый кадр из маркера, который она захватила.
Передача кадра. После удаления полей FC и ED маркера станция начинает передавать символы кадров, которые ей предоставил для передачи уровень LLC, и может передавать кадры до тех пор, пока не истечет время удержания маркера. Для передач кадров в сетях FDD1 предусмотрены два типа трафика — синхронный и асинхронный. Синхронный трафик предназначен для приложений, требующих обеспечения гарантированной пропускной способности для передачи голоса, видеоизображений, управления процессами и других случаев работы в реальном времени. Для такого трафика каждой станции предоставляется фиксированная часть пропускной способности кольца FDD1, поэтому станция имеет право передавать кадры синхронного трафика всегда, когда она получает маркер от предыдущей станции. Асинхронный трафик — это обычный трафик для приложений без высоких требований к задержкам обслуживания. Станция может передавать асинхронные кадры только в том случае, если осталось неизрасходованное время удержания маркера. Каждая станция самостоятельно вычисляет текущее значение этого параметра по специальному алгоритму. Станция прекращает передачу кадров в двух случаях: либо по истечении времени удержания маркера THT, либо после передачи всех имеющихся у нее кадров до истечения этого срока. После передачи последнего своего кадра станция формирует маркер и передает его следующей станции.
Обработка кадра станцией назначения. Распознав свой адрес в поле DA, станция назначения начинает копировать символы кадра во внутренний буфер одновременно с повторением их на выходном порту. При этом станция назначения устанавливает признак распознавания адреса. Если же кадр скопирован во внутренний буфер, то устанавливается и признак копирования (невыполнение копирования может произойти, например из-за переполнения внутреннего буфера). Устанавливается также и признак ошибки, если ее обнаружила проверка с использованием контрольной последовательности.
Удаление кадра из кольца. Каждый MAC-узел ответственен за удаление из кольца кадров, которые он ранее в него поместил. Если MAC-узел при получении своего кадра занят передачей следующих кадров, то он

удаляет все символы вернувшегося по кольцу кадра. Если же MAC-узел уже освободил маркер, то он прежде чем распознает свой адрес в поле SA повторяет на выходе несколько полей этого кадра. В этом случае в кольце возникает усеченный кадр, у которого после поля SA следуют символы Idle и отсутствует конечный ограничитель. Этот усеченный кадр будет удален из кольца какой-нибудь станцией, принявшей его, находясь в состоянии собственной передачи.

Инициализация кольца

Процедура инициализации кольца (ClaimToken) выполняется для того, чтобы все станции кольца убедились в его потенциальной работоспособности. Кроме того, в ходе этой процедуры станции должны прийти к соглашению о значении параметра T_Opr — максимально допустимого времени оборота маркера по кольцу, на основании чего они вычисляют время удержания маркера THT.
Процедура Claim Token выполняется в следующих случаях:

- при включении новой станции в кольцо и при выходе станции из кольца;
- при обнаружении какой-либо станцией факта утери маркера (маркер считается утерянным, если станция не наблюдает его в течение двух периодов максимального времени оборота маркера TOpr);
- при обнаружении длительного отсутствия активности в кольце, т. е. когда станция в течение определенного времени не наблюдает проходящих через нее кадров данных;
- по команде от блока управления станцией SMT.

Для выполнения процедуры инициализации каждая станция должна знать о своих требованиях к максимальному времени оборота маркера по кольцу. Эти требования содержатся в параметре TTRT (Target Token Rotation Time) — требуемом времени оборота маркера.
Параметр TTRT отражает степень потребности станции в пропускной способности кольца — чем меньше время TTRT, тем чаще станция желает получать маркер для передачи своих кадров. Процедура инициализации позволяет станциям узнать требования к времени оборота маркера других станций и выбрать минимальное его значение в качестве общего параметра T_Opr, на основании которого в дальнейшем будет распределяться пропускная способность кольца. Параметр TTRT, который должен находиться в пределах от 4 до 165 мс, может изменяться администратором сети.
Если какая-либо станция решает начать процесс инициализации кольца по своей инициативе, то она формирует кадр Claim Token со своим значением требуемого времени оборота маркера. Захвата маркера для этого не требуется. При этом любая другая станция, получив кадр Claim Token, начинает выполнять процедуру Claim Token. Для выполнения процедуры инициализации каждая станция поддерживает таймер текущего времени оборота маркера — TRT (Token Rotation Timer), который используется и в дальнейшем при работе кольца в нормальном режиме. TRT запускается каждой станцией при обнаружении начала процедуры Claim Token. В качестве предельного значения таймера выбирается максимально допустимое время оборота маркера. Истечение значения TRT до завершения процедуры означает ее неудачное окончание — кольцо не удалось инициализировать. В случае неудачи процесса Claim Token запускаются процедуры, с помощью которых станции пытаются выявить некорректно работающую часть кольца и отключить ее.
Процедура Clime Token работает следующим образом. Каждая станция генерирует кадр Clime со своим значением T Req, равным значению ее параметра TTRT. При этом станция устанавливает значение TOpr, равное значению TTRT. Приняв кадр Claim от предыдущей станции, она обязана сравнить значение TReq, указанное в этом кадре, со своим значением TTRT. Если другая станция просит установить время оборота маркера меньше, чем это значение (т.е. T_Req << TTRT), то данная станция перестает генерировать собственные кадры Claim и начинает повторять чужие кадры Claim, так как видит, что в кольце есть более требовательные станции. Одновременно эта станция фиксирует в своей переменной T_Opr минимальное значение TReq, которое ей встретилось в чужих кадрах Claim. Если же поступивший кадр имеет значение T_Req большее, чем собственное значение TTRT станции, то он удаляется из кольца.
Процесс Claim завершается для станции в том случае, если она получает кадр Claim со своим адресом назначения. Это означает, что данная станция является победителем состязательного процесса, и ее значение TTRT оказалось минимальным. При равных значениях параметра TTRT преимущество отдается станции с большим значением MAC-адреса.
Обнаружив, что оказалась победителем процесса Claim Token, станция должна сформировать маркер и отправить его по кольцу. Первый оборот маркера является служебным, так как за время этого оборота станции кольца узнают, что процесс Claim Token успешно завершен. При этом они устанавливают признак Ring_Operational в состояние True, означающее начало нормальной работы кольца. При следующем проходе маркера его можно будет использовать для захвата и передачи кадров данных.

Если же у какой-либо станции во время выполнения процедур инициализации значение TRT истекло, а маркер так и не появился на ее входе, то станция начинает процесс инициализации. После нормального завершения этого процесса у всех станций кольца устанавливается одинаковое значение переменной T_Opr

Управление доступом к кольцу

Управление доступом к кольцу FDDI распределено между его станциями. Каждая станция, получив маркер, самостоятельно решает, может она его захватить или нет, а если может, то на какое время.
Если у станции есть для передачи синхронные кадры, то она всегда может захватить маркер на фиксированное время, выделенное ей администратором. Если же у станции для передачи есть лишь асинхронные кадры, то условия захвата маркера определяются следующим образом.
Станция ведет таймер текущего времени оборота маркера (TRT), а также счетчик числа опозданий маркера Late_Ct. Счетчик Late_Ct всегда обнуляется, когда маркер проходит через станцию. Если же маркер опаздывает, то TRT достигает значения T_Opr раньше очередного прибытия маркера. При этом таймер обнуляется и начинает отсчет времени заново, а счетчик LateCt увеличивается на единицу, фиксируя факт опоздания маркера. При прибытии опоздавшего маркера (при этом LateCt = 1) TRT значение не сбрасывает, а продолжает считать, накапливая время опоздания маркера. Если же маркер прибыл раньше, чем истек интервал T_Opr, то значение TRT сбрасывается в момент прибытия маркера.
Возможны следующие комбинации событий, связанных с поступлением маркера и состоянием таймера:

- момент A — маркер прибыл вовремя, так как значение TRT не достигло порога T_Opr;
- момент C — время, заданное таймером, истекло раньше, чем маркер прибыл на станцию. TRT перезапускается, а счетчик LateCt увеличивается на единицу;
- момент D — маркер прибыл, но опоздал, при этом счетчик LateCt равен 1. Счетчик сбрасывается в нуль, но таймер не перезапускается, так как при поступлении маркера счетчик не был равен, нулю;
- момент E — маркер прибыл на станцию до истечения времени, заданного таймером, и при нулевом значении счетчика LateCt, поэтому считается, что он прибыл вовремя. Таймер перезапускается.

Станция может захватывать маркер только в случае, если он прибывает вовремя, т.е. если в момент его прибытия счетчик LateCt равен нулю.
Время удержания маркера управляется таймером удержания маркера — THT (Token Holding Timer). Значение этого таймера полагается равным T_Opr — TRT, где TRT — значение таймера TRT в момент прихода маркера. Если у станции в буфере имеются кадры для передачи в момент прибытия маркера и маркер прибыл вовремя, то станция захватывает его и удерживает в течение периода, определяемого значением THT. Для отслеживания разрешенного времени удержания маркера в момент его захвата значение TRT присваивается THT, а затем TRT обнуляется и перезапускается. THT считает до границы TOpr, после чего считается, что время удержания маркера исчерпано. Станция перестает передавать кадры данных и передает маркер следующей станции.
В стандарте FDDI определены еще два механизма управления доступом к кольцу. Первый — в маркере можно задавать уровень приоритета маркера, а для каждого уровня приоритета задается свое время порога, до которого считает таймер удержания маркера THT Второй — использование особой формы маркера — сдерживающего маркера (restricted token), с помощью которого две станции могут некоторое время монопольно обмениваться данными по кольцу. И, наконец, если время, заданное TRT, истечет при значении LateCt, равном 1, то такое событие считается потерей маркера и порождает выполнение процесса реинициализации кольца Claim Token.

Билет № 44.
Системы FDDI: Типы узлов и правила их соединения. Функции MAC-подуровня.

Основное назначение протокола LLC (стандарт 802.2) — обеспечение требуемого качества услуг системы передачи данных посредством передачи своих кадров либо дейтаграммным способом, либо с помощью процедур с установлением соединения и восстановлением кадров, а также обеспечение независимости вышерасположенных уровней стека протоколов от конкретного типа физической среды канала с множественным доступом.
Протоколы сетевого уровня передают через межуровневый интерфейс данные для протокола LLC: свой пакет (например, пакет IP, IPX или NetBEUI), адресную информацию об узле назначения, а также требования к качеству транспортных услуг, которое протокол LLC должен обеспечить. Протокол LLC помещает пакет протокола верхнего уровня в свой кадр, который дополняется необходимыми служебными полями. Далее через межуровневый интерфейс протокол LLC передает свой кадр вместе с адресной информацией об узле назначения соответствующему протоколу уровня MAC, который упаковывает кадр LLC в свой кадр (например, кадр Ethernet). Сначала подуровень LLC в технологиях фирм-изготовителей сетевого оборудования не выделялся в самостоятельный подуровень, и его функции растворялись в общих функциях протокола канального уровня.
В соответствии со стандартом 802.2 уровень управления логическим каналом LLC предоставляет верхним уровням три типа процедур:

- LLC1 — процедура без установления соединения и без подтверждения;
- LLC2 — процедура с установлением соединения и подтверждением;
- LLC3 — процедура без установления соединения, но с подтверждением.

Процедура без установления соединения и без подтверждения (LLC1) позволяет пользователю передавать данные с минимальными издержками благодаря дейтаграммному режиму работы. Обычно процедуры этого типа используются, когда такие функции, как восстановление данных после ошибок и упорядочивание данных, выполняются протоколами вышерасположенных уровней, поэтому не требуется дублировать их на уровне LLC.
Процедура с установлением соединения и подтверждением (LLC2) обеспечивает пользователю возможность установки логического соединения перед началом передачи любого блока данных, а также, если это требуется, позволяет выполнить процедуры восстановления после ошибок и упорядочивание потока этих блоков в рамках установленного соединения. Протокол LLC2 во многом аналогичен протоколам семейства HDLC, применяемым в глобальных сетях для обеспечения надежной передачи кадров на зашумленных линиях. Протокол LLC2 работает в режиме скользящего окна. В некоторых случаях (например, при использовании сетей в системах реального времени управляющих промышленными объектами), когда временные издержки установления логического соединения перед отправкой данных неприемлемы, а подтверждение о корректности приема переданных данных необходимо, базовая процедура без установления соединения и без подтверждения не подходит.
Процедура без установления соединения, но с подтверждением (LLC3). Выбор одного из трех режимов работы уровня LLC зависит от стратегии разработчиков конкретного стека протоколов. Например, в стеке TCP/IP уровень LLC всегда работает в режиме LLC1, выполняя простую работу извлечения из кадра и демультиплексирования пакетов различных протоколов — IP, ARP, RARP

Структура кадров LLC. Процедура с восстановлением кадров LLC2

Все кадры уровня LLC по своему назначению подразделяются на три типа: информационные, управляющие и ненумерованные.
Информационные кадры (Information) предназначены для передачи информации в процедурах с установлением логического соединения LLC2 и должны обязательно содержать поле информации. В процессе передачи информационных блоков осуществляется их нумерация в режиме скользящего окна.
Управляющие кадры (Supervisory) предназначены для передачи команд и ответов в процедурах с установлением логического соединения LLC2, в том числе запросов на повторную передачу искаженных информационных блоков.
Ненумерованные кадры (Unnumbered) предназначены для передачи ненумерованных команд и ответов, выполняющих в процедурах без установления логического соединения передачу информации, идентификацию и тестирование LLC-уровня, а в процедурах с установлением логического соединения LLC2 — установление и разъединение логического соединения, а также информирование об ошибках.

Все типы кадров уровня LLC имеют единый формат
Кадр LLC обрамляется двумя однобайтовыми полями Флаг, имеющими значение 01111110. Флаги используются на уровне MA(для определения границ кадра LLC. В соответствии с многоуровневой структурой протоколов стандартов IEEE 802, кадр LLC вкладывается в кадр уровня MAC: в кадр Ethernet, Token Ring, FDDI и т. д. При этом флаги кадра LLC отбрасываются.
Кадр LLC содержит поле данных и заголовок, включающий в себя три поля:

- адрес точки входа службы назначения (Destination Service Access Point - DSAP);
- адрес точки входа службы источника (Source Service Access Point - SSAP);
- управляющее поле (Control).

Поле данных кадра LLC предназначено для передачи по сети пакетов протоколов вышерасположенных уровней, например сетевых протоколов IP, IPX, AppleTalk. Поле данных может отсутствовать в управляющих и некоторых ненумерованных кадрах.
Адресные поля DSAP и SSAP, занимающие по одному байту, позволяют указать, какая служба верхнего уровня пересылает данные с помощью этого кадра. Программному обеспечению узлов сети при получении кадров канального уровня необходимо распознать, какой протокол вложил свой пакет в поле данных поступившего кадра, чтобы передать извлеченный из кадра пакет соответствующему протоколу верхнего уровня для последующей обработки.

Флаг 01111110	Адрес точки входа службы назначения (DSAP)	Адрес точки входа службы источника (SSAP)	Управляющее поле (Control)	Данные (Data)	Флаг 01111110

Рис. 4.19. Формат кадров уровня LLC

Для идентификации этих протоколов вводятся так называемые адреса точки входа службы (Service Access Point — SAP). Значения адресов SAP приписываются протоколам в соответствии со стандартом 802.2. Например, для протокола IP значение SAP равно 0x6, а для протокола NetBIOS — **0xF0,** Для одних служб определена только одна точка входа и соответственно только один SAP, когда адреса DSAP и SSAP совпадают, а для других — несколько. Однако иногда в кадре LLC указываются различающиеся DSAP и SSAP, что возможно только в тех случаях, когда служба имеет несколько адресов SAP, и это может быть использовано протоколом узла отправителя в специальных целях, например для уведомления узла получателя о переходе протокола отправителя в некоторый специфический режим работы.
Поле управления (1 или 2 байт) имеет сложную структуру при работе в режиме LLC2 и достаточно простую структуру при работе в режиме LLC1. В режиме LLC1 используется только один тип кадров — ненумерованный. У кадра этого типа поле управления имеет длину 1 байт. Все подполя поля управления ненумерованных кадров принимают нулевые значения, так что значимыми остаются только первые два бита поля, используемые как признак типа кадра. Так как в протоколе Ethernet при записи реализован обратный порядок битов в байте, то запись поля управления кадра LLC1, вложенного в кадр протокола Ethernet, имеет значение 0x03 (здесь и далее префикс 0x обозначает шестнадцатеричное представление). В режиме LLC2 используются все три типа кадров, и все кадры делятся на команды и ответы на эти команды. Бит P/F (Poll/Final) имеет следующее значение: в командах он называется битом Poll и требует, чтобы на команду был дан ответ, а в ответах он называется битом Final и говорит о том, что ответ состоит из одного кадра. Ненумерованные кадры используются на начальной стадии взаимодействия двух узлов, т. е. стадии установления соединения по протоколу LLC2. Поле M ненумерованных кадров определяет несколько типов команд, которыми пользуются два узла на этапе установления соединения. Приведем примеры таких команд:

- **Установить сбалансированный асинхронный расширенный режим** (SABME). Эта команда является запросом на установление соединения и одной из полного набора команд такого рода протокола HDLC. Расширенный режим означает использование двухбайтовых полей управления для кадров двух других типов;
- **Ненумерованное подтверждение** (UA). Эта команда служит для подтверждения установления или разрыва соединения;
- **Сброс соединения** (REST). Эта команда является запросом на разрыв соединения.

После установления соединения данные и положительные квитанции на кадры начинают передаваться в информационных кадрах.
Логический канал протокола LLC2 является дуплексным, следовательно, данные могут передаваться в обоих направлениях. Если поток дуплексный, то положительные квитанции на кадры также доставляются в информационных кадрах. Если же потока кадров в обратном направлении нет или необходимо передать отрицательную квитанцию, то используются супервизорные кадры.
В информационных кадрах имеется поле N(S) для указания номера отправленного кадра, а также поле N(R) для указания номера кадра, который приемник ожидает получить от передатчика следующим. При работе протокола LLC2 используется скользящее окно размером в 127 кадров, а для их нумерации циклически используется 128 чисел (от 0 до 127).
Приемник всегда помнит номер последнего кадра, принятого от передатчика, и поддерживает переменную с указанным номером кадра, который он ожидает принять от передатчика следующим. Обозначим этот номер V(R) — это и будет значение, которое передается в поле N(R) кадра, посылаемого передатчику. Если в ответ на этот кадр приемник принимает кадр, в котором номер посланного кадра N(S) совпадает с номером ожидаемого кадра V(R), то такой кадр считается корректным (если, конечно, корректна его контрольная сумма). Если приемник принимает кадр с номером N(S), который не равен V(R), то этот кадр отбрасывается и посылается отрицательная квитанция **Отказ** (REJ) с номером V(R). При приеме отрицательной квитанции передатчик обязан повторить передачу кадра с номером V(R), а также всех кадров с большими номерами, которые он уже успел отослать, пользуясь механизмом окна в 127 кадров. В состав супервизорных кадров входят следующие кадры:

- **Отказ** (REJect);
- **Приемник не готов** (Receiver Not Ready — RNR);
- **Приемник готов** (Receiver Ready — RR).

Команда RR с номером N(R) часто используется как положительная квитанция, когда поток данных от приемника к передатчику отсутствует, а команда RNR — для замедления потока кадров, поступающих в приемник. Такое замедление может быть необходимо, если приемник не успевает обработать поток кадров, присылаемых ему с большой скоростью за счет механизма окна. Получение кадра RNR требует от передатчика полной приостановки передачи, до получения кадра RR. С помощью этих кадров осуществляется управление потоком данных, что особенно важно для коммутируемых сетей, в которых нет разделяемой среды, автоматически тормозящей работу передатчика за счет того, что новый кадр нельзя передать, пока приемник LLC закончил прием предыдущего.
Итак, можно сделать следующие выводы:

- протокол LLC обеспечивает для технологий локальных сетей требуемое качество транспортной службы, передавая свои кадры либо дейтаграммным способом, либо с помощью процедур с установлением соединения и восстановлением кадров;
- предоставляет верхним уровням три типа процедур, т.е. процедуру без установления соединения и без подтверждения, процедуру с установлением соединения и подтверждением и процедуру без установления соединения, но с подтверждением;
- обеспечивает дуплексный канал, т. е. данные могут передаваться в обоих направлениях:
- использует алгоритм скользящего окна в режиме с установлением соединения;
- может управлять потоком данных, поступающих от узлов сети с помощью управляющих кадров, что особенно важно для коммутируемых сетей, в которых нет разделяемой среды, автоматически тормозящей работу передатчика при высокой загрузке сети.

Билет № 45.
Структура кадров в протоколе IEEE802.2.

Кадр начинается с преамбулы — 7 байт вида 10101010, которая в манчестерском коде на скорости 10 МГц обеспечивает интервал времени 5,6 мкс для синхронизации приемника и передатчика. Затем следует стартовый байт 10101011, обозначающий начало передачи.

Хотя стандарт IEEE 802.3 допускает двух- и шестибайтовые адреса назначения, для 10Base используются только шестибайтовые. Нуль в старшем бите адреса получателя указывает на обычный адрес, а единица — это признак группового адреса. Групповой адрес позволяет обращаться сразу к нескольким станциям одновременно. Адрес получателя, состоящий из одних единиц, — это вещательный адрес, т.е. этот кадр должны получить все станции в сети.

Адресация обеспечивает также возможность различения локального и глобального адресов. На то, какой адрес используется, указывает 46-й бит. Если этот бит равен 1 — это локальный адрес, который устанавливает сетевой администратор, и вне данной СПД этот адрес смысла не имеет. Глобальный адрес устанавливает IEEE, гарантируя при этом, что нигде в мире нет второго такого адреса. С помощью 46 бит можно получить $7\text{-}10^{13}$ глобальных адресов.

Поле данных в кадре может занимать от 0 до 1 500 байт. Поле данных длиной 0 создает проблему для обнаружения коллизий, поэтому IEEE 802.3 предписывает, что кадр не может быть короче 64 байт. Если длина поля данных недостаточна, то поле ***Заполнение*** компенсирует нехватку длины. Этот прием называется расширением носителя.

Ограничение длины кадра связано со следующей проблемой. Если кадр короткий, то станция может закончить передачу прежде, чем начало этого кадра достигнет самого отдаленного получателя. В этом случае станция может пропустить коллизию, ошибочно считая, что кадр доставлен благополучно. В IEEE 802.3 (при 2,5 км и четырех репитерах) минимальное время обнаружения коллизии равно 51,2 мкс, что соответствует 64 байт. При больших скоростях передачи длина кадра должна быть еще больше. Например, при скорости 1 Гбит и длине сегмента 2,5 км она должна будет равна 6 400 байт.

Последнее поле в структуре кадра — это контрольная сумма, которая формируется с помощью CRC-кода.

Билет № 46.
Мосты: организация, основные функции, принципы функционирования.

Довольно часто в организации возникает необходимость соединить между собой несколько ЛВС на *канальном* уровне. Почему на канальном? Потому, что чем выше мы поднимаемся по стеку протоколов вверх, тем больше затрат мы несем на обработку заголовков PDU разного уровня. Для этой цели используются специальные устройства, называемые *мостами*, которые функционируют на уровне канала данных. Это означает, что такое устройство не анализирует заголовки пакетов сетевого уровня и выше, а значит, может просто копировать кадры одной СПД в кадры другой СПД.

Рассмотрим типичные ситуации, в которых применяются мосты.

1. Многие *подразделения* в организации имеют свои *собственные разные* локальные сети, например факультеты в университетах, отделы институтах и т.п. Необходимо интегрировать информационные потоки всей организации, а, следовательно, объединить сети между собой.
2. Организация может занимать *несколько зданий*, целесообразно в каждом здании иметь свою сеть, выполнив объединение через мосты.
3. При высоких рабочих нагрузках приходится разбивать сеть на несколько *подсетей* в целях локализации трафика в каждой подсети.
4. Причиной для использования моста может служить *большое расстояние* между объединяемыми сетями, поскольку, используя мост, можно увеличить длину сегмента локальной сети.
5. Мосты, размещенные в критических точках сети, могут увеличить ее *надежность* путем блокирования узла, нарушающего работоспособность сети в целом.
6. С помощью правильно расставленных мостов можно добиться, чтобы определенный трафик проходил лишь по определенным маршрутам и не мог попасть в чужие руки. Таким образом обеспечивается *безопасность* сети.

На первый взгляд может показаться, что построить мост для каналов, работающих по стандартам IEEE 802 несложно: требуется трансформировать один формат кадра в другой. Однако, многие стандарты в семействе IEEE 802 несовместимы между собой как на физическом уровне, так и на MAC-подуровне.

Проблемы сопряжения каналов различных стандартов можно свести к следующим:

1. Каждый стандарт имеет свой *собственный формат кадра*, значит, при переходе из одной СПД в другую, требуется переформатирование кадра. Это не всегда возможно из-за отсутствия необходимой информации, кроме того, на реформатирование тратится время процессора.
2. Разные СПД могут работать *с разной скоростью*. Если передача производится из скоростной СПД в медленную СПД, то мост должен обладать достаточным буфером. Эта проблема может усугубляться непостоянством скорости передачи в результате коллизий. К тому же несколько СПД могут посылать трафик в одну и ту же СПД, что опять приведет к перекосу скоростей.
3. Мост может быть источником временной задержки, которая может влиять на *тайм-аут на верхних уровнях*. Например, сетевой уровень над канальным уровнем 802.11 пытается послать длинное сообщение в виде последовательности кадров, после отправки последнего кадра таймер устанавливается на ожидание уведомления о получении. Если сообщение проходит через мост с медленной СПД 802.3, то тайм-аут может наступить до передачи последнего кадра в медленную СПД. Сетевой уровень решит, что все сообщение утеряно, и начнет все сначала. После нескольких попыток сетевой уровень сообщит транспортному уровню, что получатель отсутствует.
4. Наиболее серьезной проблемой является то, что стандарты могут иметь *разную максимальную длину кадра*. Для 802.3 при скорости 10 Мбит/с — это 1 500 байт, а для 802.11 (WiFi) — 2 346 байт.

Билет № 47.
Прозрачные мосты (Мосты с соединяющими деревьями). Мосты с маршрутизацией от источника. Удаленные мосты.

Рассмотрим сначала *прозрачный мост*, или *мост с деревом соединений*. Основной заботой разработчиков этого моста было обеспечение его полной прозрачности, т.е. требовалось создать устройство, поддерживающее стандарт IEEE 802.3, которое пользователь мог бы купить в магазине, подключить к нему кабели своих многочисленных сегментов локальной сети и начать работать. Подключение этого устройства к СПД не должно было требовать каких-либо изменений в оборудовании, программном обеспечении, его переинсталляции, загрузки каких-либо таблиц и т. п. Как это ни удивительно, но разработчики почти достигли своей цели.

Прозрачный мост функционирует в режиме общедоступности, т. е. ему доступны все пакеты от всех сегментов СПД, подключенных к нему. По каждому поступающему кадру мост должен принять решение: надо ли его передавать дальше или сбросить, а если передавать дальше, то в какой сегмент. Для этого каждый мост должен иметь таблицу, в которой каждой станции сопоставляется номер сегмента СПД, где она находится. Эта таблица, как правило, имеющая огромные размеры, организована как таблица перемешивания, или хэш-таблица. Когда мост включают первый раз, его таблицы пусты.

Заполняются они по следующему алгоритму:

- каждый кадр с неизвестным мосту адресом доставки рассылается во все СПД, подключенные к данному мосту, кроме той, из которой поступил этот кадр;
- по реакции из каждой СПД на этот кадр мост определяет, в какой конкретно СПД находится адрес доставки и фиксирует эту информацию в таблице (как это происходит, будет рассмотрено далее).

Кадры с таким же адресом доставки будут всегда посылаться только в СПД, определенную этим алгоритмом, который называется *обучением с запаздыванием*. Топология СПД может изменяться динамически. Машины и мосты могут подключаться к СПД и исключаться из нее, поэтому для каждого элемента таблицы указывается время, когда от этой машины или моста поступал кадр. Периодически таблица просматривается, и для всех ее элементов, у которых время последнего поступления кадра отличается от текущего более чем на несколько минут, запускается процедура поиска в СПД, т.е, все изменения в СПД отслеживаются динамически. Если какую-то машину исключат из одной СПД, перенесут ее и включат в другой СПД, описанный алгоритм отметит это изменение через несколько минут.

Итак, каждый раз, когда поступает кадр, мост выполняет следующие действия:

1. Если адрес отправителя и адрес получателя один и тот же, кадр сбрасывается.
2. Если адрес отправителя и адрес получателя разные, то кадр направляется в надлежащую СПД.
3. Если в таблице нет информации о том, куда направлять кадр, его посылают во все доступные СПД.

В некоторых случаях для обеспечения большей надежности две СПД соединяются двумя мостами, однако у такой конфигурации есть следующая опасность. Пусть в СПД 1 был сгенерирован кадр **F**. Этот кадр будет сдублирован во все СПД и мостом 1, и мостом 2. Пусть мост 1 породил кадр **F1**, а мост 2 — кадр F2. Мост 1, увидев кадр **F2** с неизвестным адресом доставки, сдублирует его в СПД 1 как кадр **F3**. Также мост 2, увидев кадр **F1**, сдублирует его в виде кадра **F4**. Этот цикл будет длиться до бесконечности.

Решение данной проблемы состоит в обеспечении возможности мостам во взаимодействии друг с другом накладывать на фактическую структуру соединений СПД ограничения, в результате которых подобные циклы стали бы невозможными.

Граф соединений системы СПД, можно сократить до дерева соединений, в котором для каждой СПД в любую другую СПД имеется только один путь. Прежде всего, надо из всех мостов выбрать один в качестве корня дерева, например можно взять для этого мост с наименьшим адресом (Ethernet-адрес, который присваивают мосту при изготовлении). Адрес каждого моста уникален. Затем для полученного корня строится дерево кратчайших путей, соединяющих его с каждой СПД. В результате получают дерево соединений. Алгоритм строит единственный маршрут от корня в любую СПД. При этом хотя дерево соединений охватывает все СПД, в нем могут быть представлены не все мосты.

Прозрачные мосты хороши тем, что их достаточно только подключить, и все работает. Однако они никак не учитывают (и не могут этого делать) оптимальное распределение пропускной способности. Это привело к появлению другой схемы работы мостов — *маршрутизации от источника*.

Предположим, что отправитель знает, находится получатель в его локальной СПД или нет. Если получатель не в его локальной СПД, то отправитель устанавливает старший разряд в адресе получателя в 1. Кроме того, в заголовке кадра указывается точный маршрут, по которому будет следовать кадр. Этот маршрут представляет собой чередование 12-разрядного адреса СПД и 4-разрядного адреса моста. Номер СПД указывается после номера моста в описании маршрута.

Мост с маршрутизацией от источника ловит только те кадры, у которых старший разряд в адресе получателя равен 1. Для каждого такого кадра просматривается описание маршрута и определяется, в какую СПД отправлять этот кадр. Этот алгоритм предполагает, что каждый отправитель знает или может определить наилучший маршрут. Основная идея алгоритма поиска такого маршрута состоит в следующем. Если маршрут к получателю не известен, то отправитель посылает так называемый *поисковый кадр*, который рассылается всеми мостами по всем СПД. Когда поисковый кадр возвращается обратно, каждый мост оставляет в нем информацию о себе. Таким образом, отправитель, получив ответы на свой поисковый кадр, может выбрать наилучший маршрут среди всех присланных.

Этот алгоритм действительно позволяет найти наилучший маршрут, но он имеет один серьезный недостаток — экспоненциальный рост числа поисковых кадров. Обнаружив наилучший путь, каждый хост в СПД хранит его. Естественно, это накладывает определенные требования на ресурсы хоста, что делает использование этого подхода не столь прозрачным, как в первом случае.

Различают локальные и *удаленные мосты*. Удаленные мосты используются в больших сетях, когда ее отдельные сегменты связываются телефонными (или иными) каналами связи. В межсетевых конфигурациях удаленные мосты имеют несколько уникальных преимуществ. Одно из них связано с различием между скоростью локальной сети и скоростью глобальной сети (WAN). Хотя в настоящее время для использования в рассредоточенной межсетевой конфигурации появилось несколько быстрых технологий WAN, однако, скорость в локальной сети часто на порядок превышает скорость в WAN. Значительное различие между скоростью LAN и WAN иногда отпугивает пользователей от использования в WAN чувствительных к задержкам сетевых приложений.

Удаленные мосты скорость WAN повышать не могут, однако, они могут компенсировать расхождения в быстродействии за счет возможностей достаточной буферизации. Если устройство локальной сети со скоростью передачи 3 Mbps предполагает связь с устройством глобальной сети, локальный мост должен осуществлять регулирование потока данных со скоростью 3 Mbps так, чтобы он не переполнял последовательный канал в 64 Kbps. Это делается путем временного хранения в буферах поступающих данных с последующей их передачей по последовательному каналу с подходящей для него скоростью передачи. Такой режим возможен только для коротких пакетов данных, не перекрывающих буферные возможности моста.

Билет № 48.
Виртуальные сети на основе стандарта IEEE 802.1Q.

В виртуальных сетях, основанных на стандарте IEEE 802.1Q, информация о принадлежности передаваемых Ethernet-кадров к той или иной виртуальной сети встраивается в сам передаваемый кадр. Таким образом, стандарт IEEE 802.1 Q определяет изменения в структуре кадра Ethernet, позволяющие передавать информацию о VLAN по сети. Для этого к кадру Ethernet добавляют метку Tag длиной 4 байт и получают кадры, называемые кадрами с метками — Tagged frame, где дополнительные биты содержат информацию о принадлежности кадра Ethernet к виртуальной сети и о его приоритете.
Добавляемая к кадру метка включает в себя двухбайтовое поле TPID (Tag Protocol Identifier) и двухбайтовое поле TCI (Tag Control Information). Поле TCI состоит из полей Priority (3 бита, задает восемь возможных уровней приоритета кадра), CFI (Canonical Format Indicator, 1 бит, зарезервировано для обозначения кадров СПД других типов, передаваемых по магистрали Ethernet, *для* кадров Ethernet всегда равно 0) и V1D (VLAN ID, 12 бит, идентификатор виртуальной сети).
Изменение формата кадра Ethernet приводит к тому, что сетевые устройства, не поддерживающие стандарт IEEE 802.1Q (называемые Tag-unaware), не могут работать с кадрами, в которые вставлены метки, а подавляющее большинство этих устройств (в частности, сетевые Ethernet-контроллеры конечных узлов сети) в настоящее время не поддерживают этот стандарт. Следовательно, для обеспечения совместимости с устройствами, поддерживающими стандарт IEEE 802.1Q (Tag-aware-устройств а), коммутаторы стандарта IEEE 802.10 должны поддерживать как традиционные Ethernet-кадры, т. е. кадры без меток (Untagged), так и кадры с метками (Tagged). Для обеспечения поддержки различных типов трафиков и образования внутреннего трафика коммутатора из пакетов типа Tagged кадры на принимаемом и передающем портах коммутатора должны преобразовываться в соответствии с определенными правилами.

Правила входного порта. По отношению к трафику каждый порт коммутатора может быть как входным, так и выходным. После принятия кадра входным портом коммутатора решение о его дальнейшей обработке принимается на основании определенных правил входного порта (Ingress rules). Поскольку принимаемый кадр может быть как типа Tagged, так и типа Untagged, то правила входного порта определяют, какие типы кадров должны приниматься портом, а какие отфильтровываться. По умолчанию для всех коммутаторов правилами входного порта устанавливается возможность приема кадров обоих типов.
Если правилами входного порта определено, что он может принимать кадр типа Tagged, в котором имеется информация о принадлежности к конкретной виртуальной сети (VID), то этот кадр передается без изменения. Если же этими правилами определена возможность работы с кадрами типа Untagged, в которых не содержится информация о принадлежности к виртуальной сети, то прежде всего такой кадр преобразуется входным портом коммутатора к типу Tagged.
Чтобы такое преобразование стало возможным, каждому порту коммутатора присваивается уникальный идентификатор PVID (Port VLAN Identifier), определяющий принадлежность порта к конкретной виртуальной сети внутри коммутатора (по умолчанию все порты коммутатора имеют одинаковый идентификатор PVID = 1). Для преобразования кадра типа Untagged к типу Tagged его необходимо дополнить меткой VTD. Значение поля VID входного Untagged-кадра устанавливается равным значению PVID входящего порта, т.е. все входящие Untagged-кадры автоматически приписываются к той виртуальной сети внутри коммутатора, к которой принадлежит входной порт.

Правила продвижения пакетов. После того как все входящие кадры отфильтрованы, преобразованы или оставлены без изменения в соответствии с правилами входящего порта, решение об их передаче к выходному порту основывается на определенных правилах продвижения пакетов (Forwarding Process).
Пакеты могут передаваться только между портами, ассоциированными с одной виртуальной сетью. Порты с одинаковыми идентификаторами внутри одного коммутатора ассоциируются с одной виртуальной сетью. Если же виртуальная сеть строится на базе одного коммутатора, то идентификатора порта PV1D, определяющего его принадлежность к виртуальной сети, вполне достаточно.
Стандарт IEEE 802.1Q задумывался для обеспечения построения масштабируемой инфраструктуры виртуальных сетей, включающей в себя множество коммутаторов. При этом для расширения сети за пределы одного коммутатора наличие одних идентификаторов портов недостаточно, поэтому каждый порт может быть ассоциирован с несколькими виртуальными сетями, имеющими различные идентификаторы VID.
Если адрес назначения пакета соответствует порту коммутатора, принадлежащему к той же виртуальной сети, что и сам пакет (могут совпадать VID пакета и V1D порта или VID пакета и PVID порта), то такой пакет может быть передан. Если же передаваемый кадр принадлежит к виртуальной сети, с которой выходной порт

никак не связан (VID пакета не соответствует PVID или VID порта), то кадр не может быть передан и отбрасывается.

Правила выходного порта. После того как кадры внутри коммутатора переданы на выходной порт, их дальнейшее преобразование зависит от правил выходного порта (Egress rules). Правилами выходного порта (правилом контроля метки — Tag Control) определяется, следует ли преобразовывать кадры Tagged к формату Untagged.

Каждый порт коммутатора может быть сконфигурирован как Tagged Port или Untagged Port. Если выходной порт коммутатора определен как Tagged Port, то исходящий трафик будет создаваться кадрами типа Tagged с информацией о принадлежности к виртуальной сети. Выходной порт в этом случае не изменяет тип кадров, оставляя их такими же, какими они были внутри коммутатора. К такому порту может подсоединяться только устройство, совместимое со стандартом IEEE 802.1Q, например коммутатор или сервер с сетевой картой, поддерживающей работу с виртуальными сетями данного стандарта.

Если же выходной порт коммутатора определен как Untagged Port, то все исходящие кадры преобразуются к типу Untagged, т.е. из них удаляется дополнительная информация о принадлежности к виртуальной сети. К такому порту можно подключать любое сетевое устройство.

Билет № 49.
Сетевые коммутаторы.

Сетевой коммутатор (далее просто коммутатор) представляет собой устройство с несколькими портами, к которым можно подключать сегменты каналов с множественным доступом (далее КМД), например сегменты 802.3. На основании таблицы коммутации, расположенной в памяти коммутатора, кадры с входного порта коммутатор передает на надлежащий выходной порт, поэтому трафик на отдельных сегментах существенно ниже, чем на коммутаторе в целом. Коммутаторы работают на более высоких скоростях, чем мосты, и функционально являются более гибкими.
По сравнению с мостами у коммутаторов больше портов. Достаточно распространены коммутаторы с 24 и 48 портами со скоростями соответственно 10 и 100 Мбит/с. Коммутаторы на крупных предприятиях могут поддерживать сотни портов. У коммутаторов больше размер буферного пространства для сохранения принимаемых кадров, что весьма полезно, особенно если элементы сети перегружены. В зависимости от стоимости коммутатора возможна поддержка локальных сегментов СПД с КМД с разными скоростями: 10 Мбит/с, 100 Мбит/с, 1 Гбит/с или 10 Гбит/с.

Для коммутации данных между сетевыми портами коммутаторы используют один из следующих методов:

- *коммутация без буферизации кадров.* При использовании этого метода коммутатор, получив 6-байтовый MAC-адрес получателя, сразу занимает требуемый выходной порт и начинает передачу. Обнаружения ошибок при этом не происходит. Если на входе произойдет коллизия, то кадр будет потерян;
- *коммутация с буферизацией кадров.* При использовании этого метода коммутатор сохраняет в буфере весь кадр, причем во время сохранения кадра коммутатор анализирует его и производит обнаружение ошибок. После этого, убедившись в отсутствии коллизии на входе, он передает кадр;
- *коммутация с исключением фрагментов.* Коммутация без буферизации кадров обеспечивает малое время задержки. Получив 64-байтовый заголовок кадра, коммутатор занимает требуемый выходной порт и начинает передачу. Обнаружения ошибок при этом не происходит. Коммутация с исключением фрагментов гарантирует, что из источника считывается достаточное число байтов, чтобы обнаружить коллизию до пересылки.

Коммутация с исключением фрагментов — это метод, обеспечивающий компромисс между большим временем задержки с гарантией передачи кадра при коммутации с буферизацией, и небольшим временем задержки с риском потери кадра при коммутации без буферизации кадров. На практике разница между коммутацией с буферизацией и без буферизации кадров неважна, поскольку несущественное уменьшение времени задержки при коммутации без буферизации, возмещается незначительными колебаниями времени ожидания при коммутации с буферизацией.
Для каждого порта коммутатор формирует таблицу MAC-адресов, связанных с сегментом, подключенным к порту коммутатора. Затем коммутатор использует эти MAC-адреса при принятии решения о дальнейших операциях с кадрами: фильтрация, пересылка или лавинная рассылка.
Когда на порт поступает кадр, коммутатор сравнивает MAC-адрес адресата с адресами в таблицах. Если MAC-адрес получателя кадра находится в том же сегменте сети, что и отправитель, коммутатор сбрасывает кадр. Этот процесс называется фильтрацией, и с его помощью коммутаторы могут значительно уменьшить трафик между сегментами сети. Если MAC-адрес получателя кадра находится в другом сегменте, коммутатор пересылает кадр на порт, к которому подключен соответствующий сегмент.
Если у коммутатора нет записи об адресе получателя, то он передаст кадр всем портам, кроме того порта, с которого кадр был получен. Устройство-получатель отвечает на широковещательную рассылку специальным кадром по адресу отправителя. Коммутатор вводит искомый MAC-адрес получателя и номер соответствующего порта коммутатора в таблицу MAC-адресов. Теперь коммутатор может пересылать кадры между отправителем и получателем без широковещательной рассылки.
Коммутируемые СПД КМД — это самый распространенный в настоящее время тип СПД для локальных сетей. Сегодня цена за порт на коммутаторе уменьшилась на столько, что концентраторы и мосты больше не рассматриваются при принятии решения о покупке сетевого оборудования. Коммутаторы позволяют структурировать трафик, т. е. разбивать его на фрагменты по определенному признаку, чаще всего по физическому расположению абонентских машин пользователей. Такая организация позволяет каждой группе обращаться к устройствам в сети, например серверам, с меньшей вероятностью возникновения коллизий и повышает общую производительность сети.

Билет № 50.
Сравнение мостов и сетевых коммутаторов.

Во многом аналогичные мостам сетевые коммутаторы обладают и особыми характеристиками, которые делают их эффективным средством снижения перегрузки сетей за счет увеличения их фактической полосы пропускания.

Сходство мостов и коммутаторов заключается в следующем:

- мосты, и коммутаторы соединяют сегменты СПД КМД;
- мосты и коммутаторы используют таблицу MAC-адрссов для идентификации сегмента, в который нужно переслать кадр с данными;
- мосты и коммутаторы помогают уменьшить сетевой трафик.

Дополнительные преимущества по устранению коллизий обеспечивают следующие особые характеристики коммутаторов:

- *выделенный канал связи между устройствами.* Если на каждый порт сетевого коммутатора подключить только линию от одного абонента (это так называемая микросегментация), то каждый пользователь получит доступ к каналу передачи и не будет конкурировать с другими пользователями, а значит, и коллизий возникать не будет;
- *параллельные сеансы связи.* Параллельность сеансов связи обеспечивается посредством одновременной пересылки нескольких кадров между разными парами портов, что увеличивает пропускную способность СПД в соответствии с числом поддерживаемых сеансов связи;
- *полнодуплексная система связи.* После выполнения микросегментации в подключении участвуют только коммутатор и подключенный абонент. Теперь можно настроить порт таким образом, чтобы он мог получать и отправлять данные в одно и то же время, т.е. обеспечивать дуплексный канал связи. Например, соединение типа ≪точка—точка≫ обладает скоростью передачи 100 Мбит/с и скоростью приема 100 Мбит/с, обеспечивая эффективную пропускную способность 200 Мбит/с на одном соединении. Выбор между полудуплексом и дуплексом происходит автоматически во время создания канала;

адаптация к скорости среды. Сетевой коммутатор, имеющий порты на разные скорости, может автоматически выбирать между скоростями передачи 10/100 Мбит/с или 100/1000 Мбит/с.

Билет № 51.
Протоколы для высокоскоростных локальных сетей (Fast Ethernet, Gigabit Ethernet).

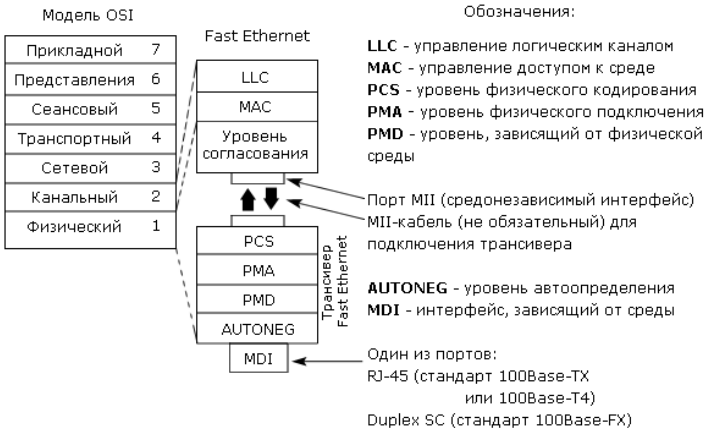
Термином *Fast Ethernet* называют набор спецификаций, разработанных комитетом IEEE 802.3, чтобы обеспечить недорогой, Ethernet-совместимый стандарт, способный обеспечить работу ЛВС на скорости 100 Мбит/сек. Необходимость в таких скоростях возникла из-за растущей диспропорции между скоростью работы процессоров рабочих станций, скоростью работы их устройств памяти и каналов ввода/вывода, в том числе и сетевых.

Отметим главные особенности эволюционного развития от сетей Ethernet к сетям Fast Ethernet стандарта IEEE 802.3u:

- десятикратное увеличение пропускной способности сегментов сети
- сохранение метода случайного доступа CSMA/CD, принятого в Ethernet
- сохранение формата кадра, принятого в Ethernet
- поддержка традиционных сред передачи данных - витой пары и волоконно-оптического кабеля

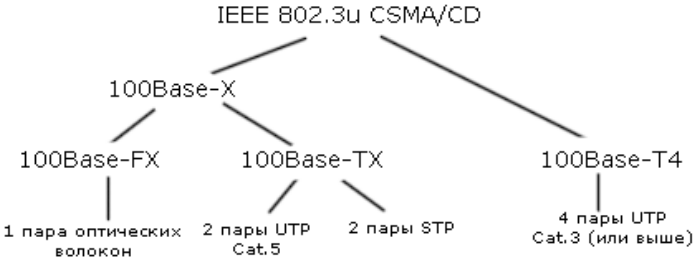
Кроме указанных свойств, важной функцией этого стандарта является поддержка двух скоростей передачи 10/100 Мбит/сек. и автоматический выбор одной из них, встраиваемая в сетевые карты и коммутаторы Fast Ethernet. Все это позволяет осуществлять плавный переход от сетей Ethernet к более скоростным сетям Fast Ethernet, обеспечивая выгодную преемственность по сравнению с другими технологиями. Еще один дополнительный фактор - низкая стоимость оборудования Fast Ethernet.

Структура уровней стандарта Fast Ethernet, MII-интерфейс и трансивер Fast Ethernet



Интерфейс MII (medium independent interface) в стандарте Fast Ethernet является аналогом интерфейса AUI в стандарте Ethernet. MII-интерфейс обеспечивает связь между подуровнями согласования и физического кодирования. Основное его назначение - упростить использование разных типов среды.
Стандартом Fast Ethernet IEEE 802.3u установлены три типа физического интерфейса: 100Base-FX, 100Base-TX и 100Base-T4.

Физические интерфейсы стандарта Fast Ethernet



100Base-FX
Стандарт этого волоконно-оптического интерфейса полностью идентичен стандарту FDDI PMD. Интерфейс Duplex SC допускает дуплексный канал связи.

100Base-TX
Стандарт этого физического интерфейса предполагает использование неэкранированной витой пары категории не ниже 5. Он полностью идентичен стандарту FDDI UTP PMD. Порт RJ-45 на сетевой карте и на

коммутаторе может поддерживать наряду с режимом 100Base-TX режим 10Base-T, или функцию автоопределения скорости. Большинство современных сетевых карт и коммутаторов поддерживают эту функцию по портам RJ-45 и, кроме этого, могут работать в дуплексном режиме.

100Base-T4

Этот тип интерфейса позволяет обеспечить полудуплексный канал связи по витой паре UTP Cat.3 и выше. Именно возможность перехода предприятия со стандарта Ethernet на стандарт Fast Ethernet без радикальной замены существующей кабельной системы на основе UTP Cat.3 следует считать главным преимуществом этого стандарта.

Интерфейс 100Base-T4 имеет один существенный недостаток - принципиальную невозможность поддержки дуплексного режима передачи. И если при строительстве небольших сетей Fast Ethernet с использованием повторителей 100Base-TX не имеет преимуществ перед 100Base-T4 (существует коллизионный домен, полоса пропускания которого не больше 100 Мбит/сек.), то при строительстве сетей с использованием коммутаторов недостаток интерфейса 100Base-T4 становится очевидным и очень серьезным. Поэтому данный интерфейс не получил столь большого распространения, как 100Base-TX и 100Base-FX.

Основные категории устройств, применяемых в Fast Ethernet, такие же, как и в Ethernet: трансиверы, конвертеры, сетевые карты (для установки на рабочие станции/файл-серверы), повторители, коммутаторы.

Трансивер - это (по аналогии с трансивером Ethernet) двухпортовое устройство, охватывающее подуровни PCS, PMA, PMD и AUTONEG, и имеющее с одной стороны MII-интерфейс, с другой - один из средозависимых физических интерфейсов (100Base-FX, 100Base-TX или 100Base-T4).

Сетевая карта. Наиболее широкое распространение сегодня получили сетевые карты с интерфейсом 100Base-TX на шину PCI. Необязательными, но крайне желательными функциями порта RJ-45 является автоконфигурирование 100/10 Мбит/сек. и поддержка дуплексного режима.

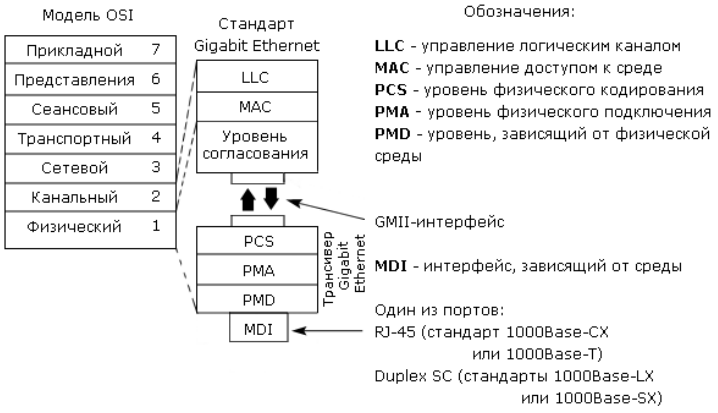
Конвертер (media converter) - это двухпортовое устройство, оба порта которого представляют средозависимые интерфейсы. Конвертеры, в отличие от повторителей, могут работать в дуплексном режиме, за исключением случая, когда имеется порт 100Base-T4.

Интерес к технологиям для локальных сетей с гигабитными скоростями повысился в связи с двумя обстоятельствами - во-первых, успехом сравнительно недорогих (по сравнению с FDDI) технологий Fast Ethernet, во-вторых, со слишком большими трудностями, испытываемыми технологией ATM на пути к конечному пользователю.

В марте 1996 года комитет IEEE 802.3 одобрил проект стандартизации *Gigabit Ethernet* 802.3z. В мае 1996 года 11 компаний организовали Gigabit Ethernet Alliance. Альянс объединил усилия большого числа ведущих производителей сетевого оборудования на пути выработки единого стандарта и выпуска совместимых продуктов Gigabit Ethernet и преследовал следующие цели:

- поддержка расширения технологий Ethernet и Fast Ethernet в ответ на потребность в более высокой скорости передачи
- разработка технических предложений с целью включения в стандарт
- выработка процедур и методов тестирования продуктов от различных поставщиков

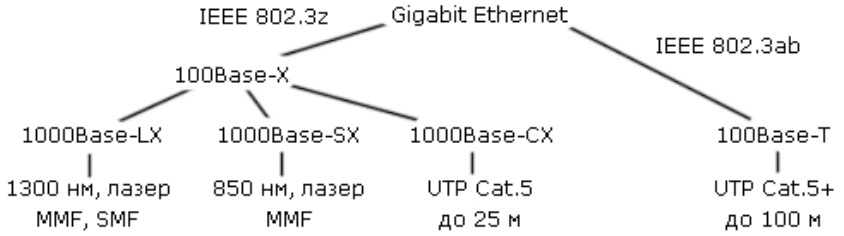
Структура уровней стандарта Gigabit Ethernet, GII-интерфейс и трансивер Gigabit Ethernet



GMII-интерфейс. Средонезависимый интерфейс GMII (gigabit media independent interface) обеспечивает взаимодействие между уровнем MAC и физическим уровнем. GMII-интерфейс является расширением интерфейса MII и может поддерживать скорости 10, 100 и 1000 Мбит/сек. Он имеет отдельные 8-битные

приемник и передатчик и может поддерживать как полудуплексный, так и дуплексный режимы. Кроме этого, GMII-интерфейс несет один сигнал, обеспечивающий синхронизацию (clock signal), и два сигнала состояния линии - первый (в состоянии ON) указывает наличие несущей, а второй (в состоянии ON) говорит об отсутствии коллизий. Также GMII-интерфейс обеспечивает еще несколько сигнальных каналов и питание. Трансиверный модуль, охватывающий физический уровень и обеспечивающий один из физических средозависимых интерфейсов, может подключаться, например, к коммутатору Gigabit Ethernet посредством GMII-интерфейса.

Физические интерфейсы стандарта Gigabit Ethernet



Интерфейс 1000Base-X основан на стандарте физического уровня Fibre Channel. Эта технология будет подробнее рассмотрена ниже. Fibre Channel - это технология взаимодействия рабочих станций, суперкомпьютеров, устройств хранения и периферийных узлов. Fibre Channel имеет 4-уровневую архитектуру. Два нижних уровня FC-0 (интерфейсы и среда) и FC-1 (кодирование/декодирование) перенесены в Gigabit Ethernet.

1000Base-X подразделяется на три физических интерфейса, различающихся характеристиками источника и приемника излучения: интерфейс *1000Base-SX* и *1000Base-LX* для многомодового оптоволокна и *1000Base-CX* для экранированной витой пары (STP «twinax») на коротких расстояниях.

1000Base-T - это стандартный интерфейс Gigabit Ethernet для передачи по неэкранированной витой паре категории 5 и выше на расстояния до 100 метров. Для такой передачи используются все четыре пары медного кабеля, скорость передачи по одной паре - 250 Мбит/сек. Предполагается, что стандарт будет обеспечивать дуплексную передачу, причем данные по каждой паре будут передаваться одновременно сразу в двух направлениях (двойной дуплекс). 1000Base-T.

Уровень MAC-стандарта Gigabit Ethernet использует тот же самый протокол передачи CSMA/CD, что и его предки Ethernet и Fast Ethernet. Основные ограничения на максимальную длину сегмента (или коллизионного домена) определяются этим протоколом. В стандарте Ethernet IEEE 802.3 принят минимальный размер кадра, равный 64 байтам. Как уже неоднократно отмечалось, именно значение минимального размера кадра определяет максимально допустимое расстояние между станциями. Время, за которое станция передает такой кадр (время канала), равно, как мы уже отмечали, 51,2 мксек. Максимальная длина сети Ethernet определяется из условия разрешения коллизий, а именно, время, за которое сигнал доходит до удаленного узла и возвращается обратно, не должно превышать 51,2 мксек. (без учета преамбулы).

При переходе от Ethernet к Fast Ethernet скорость передачи возрастает, а время трансляции кадра длины 64 байта соответственно сокращается - оно равно 5,12 мксек. Чтобы можно было обнаруживать все коллизии до конца передачи кадра, как и раньше, необходимо выполнить одно из условий:

1. Сохранить прежнюю максимальную длину сегмента, но увеличить время канала (и, следовательно, увеличить минимальную длину кадра)
2. Сохранить время канала (сохранить прежний размер кадра), но уменьшить максимальную длину сегмента

Опять же в силу преемственности, стандарт Gigabit Ethernet должен поддерживать те же минимальный и максимальный размеры кадра, которые приняты в Ethernet и Fast Ethernet. Но поскольку скорость передачи возрастает, то, соответственно, уменьшается и время передачи пакета аналогичной длины. При сохранении прежней минимальной длины кадра это привело бы к уменьшению диаметра сети, который не превышал бы 20 метров, что могло быть мало полезным. Поэтому при разработке стандарта Gigabit Ethernet было принято решение увеличить время передачи. В Gigabit Ethernet оно в 8 раз превосходит время Ethernet и Fast Ethernet. Но, чтобы поддержать совместимость со стандартами Ethernet и Fast Ethernet, минимальный размер кадра не был увеличен, зато к кадру было добавлено дополнительное поле, получившее название «расширение носителя».

Символы в дополнительном поле обычно не несут служебной информации, но они заполняют канал. В результате коллизия будет регистрироваться всеми станциями при большем диаметре коллизионного домена. Если станции нужно передать короткий (меньше 512 байт) кадр, то при передаче добавляется поле «расширение носителя», дополняющее кадр до 512 байт. Поле контрольной суммы вычисляется только для оригинального кадра и не распространяется на поле расширения. При приеме кадра поле расширения отбрасывается. Поэтому уровень LLC даже и не знает о наличии такого поля. Если размер кадра равен или превосходит 512 байт, то поле расширения носителя отсутствует.

В настоящее время поставляется полный перечень сетевых продуктов Gigabit Ethernet: сетевые карты, повторители, коммутаторы, а также маршрутизаторы. Предпочтение отдается устройствам с оптическими интерфейсами.

Билет № 52.

Сетевой уровень: проблемы построения сетевого уровня (Сервис, внутренняя организация сетевого уровня), Алгоритмы маршрутизации (принцип оптимальности, маршрутизация по наикратчайшему пути, маршрутизация лавиной, маршрутизация с анализом потока).

Основной задачей сетевого уровня является получение пакетов от всех источников и передача их по назначению. Передача по назначению может потребовать нескольких этапов, нескольких маршрутизаторов на маршруте. Для реализации своей функции сетевой уровень должен знать топологию транспортной подсети и выбрать подходящий путь в ней. Выбирая маршрут, он должен позаботиться, чтобы этот маршрут не привел к перегрузкам некоторых линий и маршрутизаторов. Наконец, если источник и получатель принадлежат разным сетям, то задача сетевого уровня, принимая во внимание различия между этими сетями, обеспечить корректную передачу данных из одной сети в другую.
Сетевой уровень предоставляет сервис транспортному через интерфейс между ними.
Сервис сетевого уровня разрабатывался в следующих целях:

- сервис должен быть независимым от технологии передачи, используемой в СПД среде;
- транспортный уровень должен быть независим от числа, типа и топологии транспортной подсети;
- адрес в СПД среде, доступный на транспортном уровне, должен иметь унифицированную форму по всей сети.

Сервис, ориентированный на соединение, предполагает, что эта сложность приходится на сетевой уровень, т.е. на транспортную среду. Сервис без соединений - на транспортный уровень, а стало быть на хост.

Внутренняя организация сетевого уровня

С точки зрения внутренней организации сетевой уровень делится на ориентированный на соединения и без соединений. В первом случае соединение называют виртуальным каналом, по аналогии с физическим каналом в телефонных сетях. Во втором случае, о пакетах говорят как о дейтаграммах, по аналогии с телеграммами. Идея виртуального канала – избежать маршрутизации для каждого пакета. Маршрут устанавливается один раз при установлении виртуального канала между отправителем и получателем и в дальнейшем не меняется до тех пор, пока передача не закончится. При подходе без соединения каждый пакет маршрутизируется независимо. Разные пакеты могут следовать разными маршрутами.
Каждый маршрутизатор в сети, ориентированной на виртуальные каналы, должен помнить какие каналы проходят через него. У каждого маршрутизатора есть таблица виртуальных каналов. Каждый пакет должен иметь дополнительное поле, где храниться номер виртуального канала. Когда пакет приходит к маршрутизатору то, зная линию, по которой он пришел, и номер виртуального канала, указанный в пакете, по таблице маршрутизатор устанавливает, по какой линии надо отправить пакет далее. При установлении соединения номер виртуального канала выбирается из числа не используемых в данный момент на данной машине.
У каждой дейтаграммы должен быть полный адрес доставки. В больших сетях этот адрес может быть достаточно большим (десятки байт). Когда пакет поступает, маршрутизатор по таблице и адресу определяет по какой линии надо отправить эту дейтаграмму и посылает ее туда.Пакеты маршрутизируются всегда, независимо от того какую внутреннюю организацию имеет транспортная среда - с виртуальными каналами или дейтаграммную. Разница лишь в том, что в первом случае этот маршрут устанавливается один раз для всех пакетов, а во втором - для каждого пакета.
Алгоритм маршрутизации - часть программного обеспечения сетевого уровня, и отвечает за определение по какой линии отправлять пакет дальше. В независимости от того выбирается ли маршрут для сессии или для каждого пакета в отдельности алгоритм маршрутизации должен обладать рядом свойств: *корректностью, простотой, устойчивостью, стабильностью, справедливостью и оптимальностью*.

Алгоритм маршрутизации должен быть ***устойчивым***, т.е. сохранять работоспособность независимо ни от каких сбоев или отказов в сети, изменений в ее топологии: отключение хостов, машин транспортной подсети, разрушения каналов и т.п.
Стабильность сходимость алгоритма к равновесному состоянию при продолжительной работе.
Справедливость значит, что все пакеты, вне зависимости от того, из какого канала они поступили, будут обслуживаться равномерно, для всех абонентов будет всегда выбираться *оптимальный маршрут*. Возможные *критерии оптимизации*:- минимизация средней задержки пакета; максимизация пропускной способности сети; минимизация числа переходов между маршрутизаторами - **скачков** (hop). Уменьшение числа скачков сокращает маршрут, следовательно, сокращает задержку, минимизирует потребляемую пропускную способность при передаче пакета.
Алгоритмы маршрутизации: адаптивные и неадаптивные.

Неадаптивные алгоритмы не принимают в расчет текущую загрузку сети и состояние топологии. Все возможные маршруты вычисляются заранее и загружаются в маршрутизаторы при загрузке сети **(статическая маршрутизация).**
Адаптивные алгоритмы определяют маршрут, исходя из текущей загрузки сети и топологии. Адаптивные алгоритмы различаются тем, где и как они получают информацию
Принцип оптимальности утверждает, что если маршрутизатор *J* находится на оптимальном пути между маршрутизаторами *I* и *K*, то оптимальный маршрут между *J* и *K* принадлежит этому оптимальному пути. Следствием из принципа оптимальности является утверждение, что все маршруты к заданной точке сети образуют дерево с корнем в этой точке – **дерево захода.** В нём нет циклов => каждый пакет будет доставлен за конечное число скачков.

Маршрутизация по наикратчайшему пути
Идея этого алгоритма состоит в построении графа транспортной среды, где вершины - маршрутизаторы, а дуги - линии связи. Алгоритм находит для любой пары маршрутизаторов, а точнее абонентов, подключенных к этим маршрутизаторам, наикратчайший маршрут в этом графе. В общем случае веса на дугах могут быть функциями от расстояния, пропускной способности канала, среднего трафика, стоимости передачи, средней длины очереди в буфере и других факторов. Изменяя весовую функцию, алгоритм будет вычислять наикратчайший путь в смысле разных мер. Известно несколько алгоритмов вычисления наикратчайшего пути в графе. Алгоритм Дейкстры.

Маршрутизация лавиной - каждый поступающий пакет отправляют по всем имеющимся линиям, за исключением той, по которой он поступил. Если ничем не ограничить число повторно генерируемых пакетов, то их число может расти не ограниченно. Для этого в заголовке каждого изначально генерируемого пакета устанавливается счетчик скачков. При каждой пересылке этот счетчик уменьшается на единицу. Когда он достигает нуля, пакет сбрасывается и далее не посылается. Другим приемом, ограничивающим рост дублируемых пакетов, является отслеживание на каждом маршрутизаторе тех пакетов, которые через него однажды проходили.

Маршрутизация на основе потока - ***статический алгоритм маршрутизации на основе потока,*** учитывает, как топологию, так и загрузку транспортной подсети.
О каждой транспортной среде необходимо знать заранее: топологию; матрицу трафика F_{ij} ; матрицу пропускных способностей C_{ij}; алгоритм маршрутизации.
Имея эти данные, не трудно построить алгоритм вычисления наикратчайшего пути с точки зрения весов на дугах графа. Если трафик измениться по какой-либо причине, то достаточно перевычислить таблицу, не меняя алгоритма.

Билет № 53.

Сетевой уровень: проблемы построения сетевого уровня (Сервис, внутренняя организация сетевого уровня). Алгоритмы маршрутизации (маршрутизация по вектору расстояния, маршрутизация по состоянию канала, иерархическая маршрутизация, маршрутизация для мобильного узла, маршрутизация при вещании, маршрутизация для группы).

Динамическую маршрутизация. Один из наиболее популярных алгоритмов - **маршрутизация по вектору расстояния,** построен на идеях алгоритмов Беллмана-Форда и Форда-Фолкерсона. Изначально использовался в сети ARPA и используется по сей день под названием RIP алгоритма. В основе его лежит идея, что у каждого маршрутизатора в транспортной подсети есть таблица расстояний до каждого маршрутизатора в подсети. Периодически маршрутизатор обменивается такой информацией со своими соседями и обновляет информацию в своей таблице. Каждый элемент таблицы состоит из двух полей: первое - номер линии, по которой надо отправлять пакеты, чтобы достичь нужного места, второе - величина задержки до места назначения. Эта величина задержки может быть измерена в разных единицах: скачках, миллисекундах, длине очереди на линии и т.д. Каждые Т секунд маршрутизатор шлет своим соседям свой вектор задержек до всех маршрутизаторов в подсети. В свою очередь он получает такие же вектора от своих соседей. Кроме этого, он постоянно замеряет задержки до своих соседей.

Проблема счетчика до бесконечности.
Алгоритм маршрутизации по вектору расстояния теоретически работает хорошо, но у него есть один недостаток: он очень медленно сходится к правильному значению. Информация о появлении хорошего маршрута в подсети распространяется более или менее быстро, а вот данные о потере, разрушении какого-то маршрута распространяются не столь быстро. А именно это зависит от значения бесконечности в данной сети. Так как в сети есть старые маршруты, которые проходят через маршрутизаторы, которые уже поняли что этот маршрут разрушен.

Разделение направлений (Split Horizon Hack)
Одним из решений этой проблемы является следующий прием. Алгоритм работает так, как было описано, но при передаче вектора по линии, по которой направляются пакеты для маршрутизатора X, т.е. по которой достижим маршрутизатор X, расстояние до X указывается как бесконечность. Чтобы не было обратного хода пакета. Проблема остаётся в случае цикла.

Маршрутизация по состоянию линии
Основная идея построения этого алгоритма проста и состоит из пяти основных шагов:
1. Определить своих соседей и их сетевые адреса;
2. Измерить задержку или оценить затраты до каждого соседа;
3. Сформировать пакет, где указаны все данные, полученные на шаге 2;
4. Послать этот пакет всем другим маршрутизаторам;
5. Вычислить наикратчайший маршрут до каждого маршрутизатора.

Определение соседей. При загрузке маршрутизатор прежде всего определяет кто его соседи. Для этого он рассылает по всем своим линиям точка-точка специальный пакет HELLO. В ответ все маршрутизаторы отвечают, указывая свое уникальное имя.
Оценка затрат. Оценка затрат до каждого соседа происходит с помощью другого специального пакета ECHO. Это пакет рассылается всем соседям и замеряется задержка от момента отправки этого пакета до момента его возвращения. Все, кто получает такой пакет, обязаны отвечать незамедлительно. Такие замеры делают несколько раз и вычисляют среднее. Если пакет обрабатывается в общей очереди, то можно оценивать загруженность канала.
Формирование пакета состояния канала. После того как измерения выполнены, можно сформировать пакет о состоянии каналов. В пакете указано кто отправитель, последовательное число, возраст список соседей и задержки до них.
Распространение пакетов состояния каналов. Как только СК пакет получен и включен в работу, маршрутизатор будет его использовать при определении маршрута. При неудачном распространении СК пакетов, разные маршрутизаторы могут получить разное представление о топологии СПД среды, что может приводить к циклам, недостижимым машинам и другим проблемам. СК пакеты распространяются методом лавины. Однако, чтобы не потерять контроль и не вызвать неограниченное дублирование СК пакетов, каждый маршрутизатор ведет счетчик последовательных номеров СК пакетов, которые он сгенерировал. Все маршрутизаторы запоминают пары (маршрутизатор, последовательное число), которые они уже встречали среди полученных СК пакетов. Если поступивший СК пакет содержит пару, которая еще не встречалась

маршрутизатору, то он отправляет этот СК пакет всем своим соседям, за исключением того, от которого он его получил. Если он уже встречал такой пакет, то пакет сбрасывается и никуда не дублируется.
У этого алгоритма есть несколько проблем, но все они разрешимые
1. Размер поля последовательных номеров пакетов
2. Если маршрутизатор «упал» по какой-либо причине и потерял последовательность чисел, то неясно, как ее восстановить.
3. Если в результате передачи возникнет ошибка в одном бите, например, вместо 4 получим пакет с номером 65540, то все пакеты с 5 по 65540 будут сбрасываться как устаревшие
Для решения этих проблем используется поле возраст СК пакета
В целях сокращения числа рассылаемых СК пакетов, когда такой пакет поступает, его не сразу дублируют и отправляют. Сначала его помещают в специальную область задержки. Там он находится некоторое время. Если за это время придет другой пакет от того же источника, то пакеты сравниваются. Если нет различий между ними, то вновь пришедший сбрасывается, если есть, то последний пришедший дублируется и отправляется другим, а первый сбрасывают. Все СК пакеты передают с уведомлением.

Иерархическая маршрутизация
По мере роста СПД среды размер таблиц, т.е. затраты памяти, время процессора на обработку этих таблиц, пропускная способность каналов, затрачиваемая на передачу служебной информации, может превысить разумные пределы. Таким образом, дальнейший рост сети, когда каждый маршрутизатор знает все о каждом другом маршрутизаторе, будет не возможен. Решение этой проблемы – иерархия сетей, подобно тому, как в телефонной сети есть иерархия коммутаторов. Однако за эту экономию приходится платить эффективностью маршрутизации. Клейнрок и Камоун показали, что оптимальное число уровней иерархии в СПД среде при N узлах будет $\ln N$, при $e \ln N$ строках в таблице маршрутизатора.

Маршрутизация для мобильного узла
Как только мобильный узел подключается к местной, локальной сети, он регистрируется у агента визитеров. Эта процедура примерно выглядит так:
1. Периодически агент визитеров рассылает по своей области пакет, где указано место расположения этого агента и его адрес. Если мобильный узел, подключившись к сети долго не видит такого пакета, он рассылает свой пакет с просьбой агенту визитеров объявить свои координаты.
2. Мобильный узел регистрируется у агента визитеров, указывая свое текущее место положение, домашнее местоположение и определенную информацию, связанную с безопасностью передаваемых данных(пароль и тип соеденения).
3. Агент визитеров обращается через сеть к домашнему агенту домашнего местоположения визитера, указывая, что один из его пользователей сейчас находится в его области, передавая конфиденциальную информацию, которая должна убедить домашнего агента, что это действительно его пользователь пытается соединиться с ним.
4. Домашний агент изучает конфиденциальный данные, время связи. Если эти данные соответствуют той инофрмации, что есть у домашнего агента об этом пользователе, он дает добро на связь.
5. Агент визитеров, получив подтверждение от домашнего агента, заносит данные о мобильном узле в свои таблицы и регистрирует его.
Рассмотрим теперь что происходит, когда кто-то посылает сообщения мобильному узлу. Пакет поступает на адрес домашнего местоположения пользователя, где его перехватывает домашний агент. Домашний агент инкапсулирует этот пакет в свой пакет, который он отправляет по адресу агента визитеров той области, откуда последний раз был сеанс связи с пользователем. Одновременно с эти домашний агент посылает сообщение отправителю пакета, чтобы он все последующие пакеты мобильному узлу инкапсулировал в сообщениях, направляемых по адресу агента визитеров. Такой механизм инкапсулирования одних пакетов в другие называется ***тунелированием*** и мы его подробно рассмотрим позднее.

Маршрутизация при вещании
Есть несколько способов реализации такого режима.
Первый, источник знает, кому надо послать, и генерирует столько сообщений, сколько получателей. Это одно из самых плохих решений.
Метод лавины – другое решение. Однако, как мы уже видели, он затратен для каналов точка-точка.
Третий подход – маршрутизация множественной доставки. Здесь каждый пакет должен иметь либо лист рассылки, либо карту рассылки. Каждый маршрутизатор, получив такой пакет, отправляет и дублирует его в соответствии с картой рассылки.

Четвертый поход основан на использовании дерева захода либо любое другое подходящее дерево связей. Дерево захода позволяет избежать циклов и ненужного дублирования пакетов. Каждый маршрутизатор дублирует пакет вдоль линий, соответствующих дереву захода, кроме той, по которой пакет пришел.
Пятый подход основан на неявном использовании дерева связей. Когда пакет поступает, маршрутизатор проверяет, если он поступил по линии, которая используется для отправления пакетов источнику вещательного пакета, то вещательный пакет дублируется и рассылается по всем линиям, кроме той, по которой пакет пришел. Если нет, то он сбрасывается. Это метод называется **пересылка вдоль обратного пути.**

Маршрутизация при групповой передаче
Этот вид передачи используют, когда надо обеспечить взаимодействие группы взаимосвязанных процессов, разбросанных по сети. Если группа велика в сравнении с размерами сети, то можно воспользоваться методами вещания (вещательные адреса и тд). Алгоритм групповой маршрутизации, как правило, основан на дереве связей. Каждый маршрутизатор в СПД вычисляет дерево связей, охватывающее все другие маршрутизаторы.

Билет № 54.

Сетевой уровень: проблемы построения сетевого уровня (Сервис, внутренняя организация сетевого уровня). Алгоритмы управления перегрузками на сетевом уровне (Основные принципы управления перегрузками, методы предотвращения перегрузок: формирование трафика, спецификация потока, управление перегрузками в сетях с виртуальными каналами; методы устранения перегрузок: подавляющие пакеты, сброс трафика; управление перегрузками при вещании).

Когда в СПД среде находится в одно и тоже время слишком много пакетов, ее производительность начинает падать. Затор или перегрузка может возникнуть в силу нескольких причин. Например, если сразу несколько потоков, поступающих по нескольким входным линиям, устремятся на одну и ту же выходную линию. Если буфер маршрутизатора переполнится, то пакеты начнут теряться. Если процессор будет не в состоянии справиться своевременно с рутинными задачами, то даже при наличии линий с достаточной пропускной способность очередь будет расти. Аналогичная картина может случиться при быстром процессоре, но медленной линии и наоборот. Таким образом, *источник проблемы - несбалансированность производительности компонентов системы.*

Перегрузки имеют тенденцию к самостоятельному росту и ухудшению ситуации. **Перегрузка** - это глобальная проблема в сети. **Управление перегрузками** - это такая организация потоков в подсети, при которой потоки соответствуют пропускной способности подсети и не превышают ее. Это глобальная проблема в сети, затрагивающая поведение всех хостов, всех маршрутизаторов. **Управление потоком** возникает между парой взаимодействующих хостов. Это локальная проблема, касающаяся двух взаимодействующих хостов.

Часто управление перегрузкой и управление потоком путают из-за того, что и там и там применяют одинаковые приемы, например, направляют источникам специальные пакеты, тормозящие нарастание потоков.

Основные принципы управления перегрузками

В терминологии теории управления все методы управления перегрузками в сетях можно разбить на две большие группы: с открытым контуром управления и закрытым контуром управления. Методы с открытым контуром предполагают, что все продумано и предусмотрено заранее в конструкции системы и если нагрузка находится в заданных пределах, то перегрузки не происходит. Если же нагрузка начинает превышать определенные пределы, то заранее известно, когда и где начнется сброс пакетов, в каких точках сети начнется перепланировка ресурсов и т.п. Главное, что все эти меры будут приниматься, не обращая внимания на текущее состояние сети.

Решения, основанные на замкнутом контуре, используют обратную связь. Эти решения включают три этапа:

- наблюдение за системой, чтобы определить где и когда началась перегрузка;
- передача данных туда, где будут предприняты надлежащие меры;
- настроить функционирование системы так, чтобы устранить проблему.

При наблюдении за системой используются разные метрики для определения перегрузки. Основными среди них являются:

- процент пакетов, сброшенных из-за нехватки памяти в буферах;
- средняя длина очередей в системе;
- число пакетов, для которых наступил time_out
- средняя задержка пакета при доставке и среднее отклонение задержки при доставке пакета.

Следующий шаг при использовании обратной связи - передать информацию о перегрузке туда, где что-то может быть сделано, чтобы исправить положение. Маршрутизатор, обнаруживший перегрузку, направляет сообщение о перегрузке всем источникам сообщений.

Другое решение в сети рассылаются специальные пробные пакеты, которые проверяют нагрузку и если где-то обнаружена перегрузка, то о ней сообщается всем и происходит перенаправление пакетов так, чтобы обогнуть перегруженные участки.

Решения с открытым контуром, в свою очередь, делятся на две группы: воздействующие на источники и воздействующие на получателей. Решения с закрытым контуром - с явной обратной связью и не явной обратной связью. Явная обратная связь предполагает, что источнику посылается специальный пакет, который информирует его о перегрузке. Не явная обратная связь основана на том, что источник сам определяет факт перегрузки на основе своих локальных наблюдений за трафиком.

Появление перегрузки означает, что нагрузка превысила, возможно временно, ресурсы системы или некоторой ее части. Есть два выхода из этого положения: увеличить ресурсы и сократить нагрузку.

Методы, предотвращающие перегрузки
Идея в том что со всем следующим можно бороться путём правильной настройки. Для систем с открытым контуром. Эти методы ориентированы на минимизацию перегрузок при первых признаках их проявлений, а не на борьбу с перегрузками, когда они уже случились. Начнем с канального уровня. Вызвать перегрузку может повторная пересылка кадров. Организация рассылки уведомлений так же влияет на перегрузку. На сетевом уровне выбор схемы работы: с виртуальными соединениями или дейтаграммами, влияет на появление перегрузок. Выбор метода сброса пакетов также влияет на перегрузки. Правильная маршрутизация, равномерно использующая каналы в СПД сети, позволяет избежать перегрузки. Методы, регулирующие время жизни пакета в сети, так же влияют на образование перегрузок. На транспортном уровне возникают те же самые проблемы, что и на канальном.

Формирование трафика
Одной из основных причин перегрузки является не регулярный, взрывообразный трафик.
Формирование трафика регулирует среднюю скорость передачи данных и предотвращает тем самым его взрывообразность. Когда пользователь и СПД среда договариваются о форме трафика, то они приходят к соглашению не только о форме трафика, но также и о том, что произойдет, если эта форма будет нарушена пользователем. Это соглашение называется ***соглашение о трафике.***

Алгоритм текущего ведра
Идея этого алгоритма показана ведро может наполняться с любой скоростью, но вытекать из него пакеты будут со строго определенной скоростью, при переполнение его пакеты сбрасываются. В качестве регулятора скорости поступления пакетов можно использовать системные часы.

Алгоритм ведра с маркерами
Идея его заключается в том, что вместе с пакетами в ведро поступают маркеры. Пакеты из ведра уходят в сеть только при наличии соответствующего количества маркеров. Таким образом, можно накапливать маркеры и кратковременно ускорят передачу пакетов в сеть. Другое отличие алгоритма ведра с маркером - при переполнении буфера хосту будет временно запрещено передавать пакеты(нет маркеров). Для реализации алгоритма ведра с маркерами нужна лишь переменная, значение которой увеличивается каждые ΔT сек., и уменьшается с каждым посланным пакетом.
Длительность всплеска при передачи на основе уравнения
$C + \rho S = MS$, *где* C – объем буфера, S – длительность всплеска, ρ – скорость поступления маркера, M – максимальная скорость “вытекания” данных. Таким образом $S = C/(M - \rho)$.

Спецификация потока
Формирование трафика эффективно тогда, когда отправитель, получатель и среда передачи заранее договорились о его форме. Это соглашение называется *спецификацией потока*. Оно состоит из структуры данных, которая определяет как форму выходного трафика, так и качество сервиса, необходимого приложению. Эта спецификация применима как к пакетам, передаваемым по виртуальным каналам, так и к дейтаграммам.

Управление перегрузками в сетях с виртуальными каналами
До сих пор мы рассматривали методы управления перегрузками, основанные на открытом контуре. Мы здесь рассмотрим только один метод устранения уже возникшей перегрузки в сетях с виртуальными каналами - динамический контроль доступа. Прием, который широко используется, чтобы сдержать уже возникшую перегрузку и не дать положению ухудшиться - это контроль на входе. Идея очень проста - если обнаружена перегрузка, то все что способствует увеличению трафика запрещено. Прежде всего, запрещается создание новых виртуальных соединений. Другой подход разрешает установку новых виртуальных соединений, но только при наличии не перегруженных маршрутов. Третий подход уже упоминался: хост и СПД среда договариваются перед установкой виртуального соединения о форме трафика, объеме передаваемых данных, качестве сервиса, если эти соглашения нарушены, трафик может не пройти.

Подавляющие пакеты
Теперь давайте рассмотрим приемы, используемые как в средах с виртуальными каналами, так и в средах с дейтаграммами. Каждый маршрутизатор может контролировать степень загрузки своих выходных линий и другие ресурсы. Всякий раз, когда степень загруженности, при очередном вычислении, оказывается выше некоторого порога, эта линия переводится состояние предупреждения. Каждый пакет, маршрутизируемый через такую линий, вызывает генерацию подавляющего пакета, направляемого отправителю

маршрутизируемого пакета. При этом, в пакете отправителя проставляется определенный разряд, предотвращающий генерацию подавляющих пакетов другими маршрутизаторами в дальнейшем. Когда отправитель получает подавляющий пакет, он сокращает интенсивность своего трафика на определенную величину.
В течение определенного времени отправитель будет игнорировать подавляющие пакеты, поступающие с направления получателя. По истечении этого периода времени отправитель ожидает появления подавляющих пакетов в течение следующего интервала. Если появился хоть один подавляющий пакет, то линия перегружена и отправитель ждет. Если в течение очередного интервала не поступило ни одного подавляющего пакета, то отправитель может увеличить интенсивность трафика.
Недостаток. Если есть несколько отправителей, работающих через одну и ту же выходную линию, то при определенных условиях маршрутизатор всем им пошлет подавляющие пакеты.
Это приведет к несправедливому использованию пропускной способности канала между ними.
Алгоритм справедливого чередования. Суть его состоит в том, что для каждого отправителя у выходной линии строится своя очередь. Отправка пакетов из этих очередей происходит по кругу. Поэтому, если кто-то из отправителей не значительно сократит трафик, то это лишь увеличит скорость роста его очереди.
Недостаток: если один отправитель использует длинные пакеты, а другой короткие, то последний получит меньшую долю пропускной способности линии. Для борьбы с этой несправедливостью в алгоритм обслуживания очередей вносят модификацию: пакеты из очередей передаются по байтно, а не весь пакет сразу.
Другие модификации алгоритма чередования связаны с установкой приоритетов между очередями, что дает большую гибкость в обслуживании отправителей.
Представленные здесь алгоритмы с подавляющими пакетами плохо работают в высокоскоростных сетях и на больших расстояниях. Дело в том, что пока подавляющий пакет дойдет до отправителя, пройдет много времени и отправитель успеет “напихать” в сеть много пакетов. Для исправления этой ситуации была предложена его модификация – алгоритм с подавлением по скачкам. Суть его в том, что как только обнаружится перегрузка на выходной линии и маршрутизатор отправит подавляющий пакет, то ближайший маршрутизатор, получивший этот подавляющий пакет, сократит трафик к маршрутизатору, пославшему подавляющий пакет. Естественно, этот прием увеличит нагрузку на буфера маршрутизаторов, сокращающих трафик,

Сброс нагрузки
Когда ни один из упомянутых выше приемов не срабатывает, маршрутизатор может применить – сброс нагрузки. Маршрутизатор может сбрасывать пакеты, исходя из информации о приложении, пославшего эти пакеты. В общем случае подход, на основе сброса нагрузки, предполагает определенное взаимодействие между приложением и маршрутизатором. Для обеспечения такого взаимодействия с приложением вводят приоритеты среди пакетов. Это позволяет маршрутизатору минимизировать потери для приложения, когда маршрутизатор вынужден сбрасывать пакеты. Приоритеты используются, например, в протоколах Frame Relay.

Управление перегрузками при групповой передаче
Все алгоритмы управления перегрузками, которые мы до сих пор рассматривали, относились к случаю, когда был один источник и один получатель. Здесь рассматривается случай управления нагрузкой при групповой передаче, т.е. когда есть несколько источников и несколько получателей.
RSVP (Resource reSerVation Protocol) – протокол резервирования ресурсов
Он позволяет нескольким отправителям передавать сообщения группам получателей, отдельным получателям переходить из группы в группу, оптимизировать использование пропускной способности каналов, избегая перегрузок. В своей простейшей форме этот протокол для групповой маршрутизации использует дерево связей так, как мы уже рассматривали ранее. Каждой группе приписана группа адресов. При отправке пакета отправитель помещает в него весь список адресов группы. После этого стандартный алгоритм групповой маршрутизации строит дерево связей, покрывающее все адреса группы. Собственно маршрутизация не является частью RSVP. Для того чтобы избежать перегрузок, любой получатель в группе шлет надлежащему отправителю резервирующее сообщение. Это сообщение с помощью алгоритма пересылки вдоль обратного пути, движется к отправителю и вызывает резервирование необходимой пропускной способности на каждом узле, через который оно проходит. Если при прохождении очередного узла ему не удается зарезервировать необходимой пропускной способности, то получателю направляется отказ в установлении соединения.

Билет № 55.

Сетевой уровень: проблемы построения сетевого уровня (Сервис, внутренняя организация сетевого уровня). Межсетевое взаимодействие (соединение виртуальных каналов, межсетевая передача без соединений, тунелирование, межсетевая маршрутизация, фрагментация, Firewall).

Рассмотрим случай, когда соединение возникает между СПД средами, с разной архитектурой. Название средства, соединяющего сети между собой, зависит от того, на каком уровне это происходит.

- Уровень 1: репитор копирует биты из одного кабельного сегмента в другой;
- Уровень 2: мост передает пакеты канального уровня из одной ЛВС в другую;
- Уровень 3: мультипротокольный маршрутизатор передает пакеты между сетями с разной архитектурой;
- Уровень 4: транспортный шлюз соединяет байтовые потоки на транспортном уровне;
- Над уровнем 4: прикладной шлюз соединяет приложения в разных сетях.

Шлюз соединяет сети с разной архитектурой. Репитор - устройство обеспечивающее усиление и очистку сигнала. На MAC уровне трансивер обеспечивает передачу сигнала в пределах 500 метров. Репитор обеспечивает передачу на 2.5 км. Мост способен хранить и маршрутизировать пакеты на канальном уровне. Он получает канальный пакет целиком и решает по какой линии его передать дальше. Мультипротокольные маршрутизаторы функционально, примерно то же самое, что и мосты, но работают на сетевом уровне. Они получают пакеты сетевого уровня и определяют, куда их передать. Однако при этом, разные каналы могут принадлежать разным сетям, использующим разные протокольные стеки. Поэтому мультипротокольному маршрутизатору кроме задачи маршрутизации приходится решать и задачу сопряжения форматов пакетов на сетевом уровне в сетях с разной архитектурой.

Сопряжение виртуальных каналов

Есть два общих приема для межсетевого взаимодействия**:** сопряжение, ориентированное на соединения подсетей с виртуальными каналами, и взаимодействие подсетей через дейтаграммы. Абонентская машина одной сети устанавливает виртуальное соединение не только внутри своей сети, но и в другой сети, вплоть до получателя. Внутри своей сети соединение прокладывается по правилам этой сети вплоть до мультипротокольного маршрутизатора, ближайшего к сети получателя. Затем от этого маршрутизатора до получателя по правилам сети получателя. Сопряжение виртуальных каналов (достоинства): буфера можно резервировать заранее, порядок пакетов сохраняется, проще управлять повторной передачей из-за задержки, короткие заголовки пакетов. Сопряжение виртуальных каналов (недостатки): хранение таблицы сопряжения, сложности в изменении маршрута при перегрузках, требование высокой надежности маршрутизаторов вдоль сопряжения.

Межсетевое взаимодействие без соединений

Решение на основе соединения сетей на уровне дейтаграмм. В этом подходе единственный сервис, какой сетевой уровень предоставляет транспортному – “впрыскивание” дейтаграмм в СПД среду. Такое сопряжение сетей возможно, если соединяемые СПД среды используют одни и те же или очень близкие сетевые протоколы. Другая проблема - адресация. Различия в адресации могут быть столь велики, что сопряжение станет не возможным. Выход - распространять каждую адресацию на все машины в мире. Другой выход - создать универсальный пакет, который понимали бы разные сети, тоже не работает. Основное достоинство дейтаграммного подхода - он может использоваться между сетями, которые не поддерживают виртуальных соединений. Категория таких сетей весьма велика.

Тунелирование

Соединение двух одинаковых сетей через третью. Решение проблемы межсетевого соединения в этом случае дает применение техники **тунелирования**. Суть этого приема состоит в том, что пакет из одной сети упаковывается в кадр промежуточной сети. Затем он передается через промежуточную сеть на канальном уровне. При достижении кадром сети назначения, кадр распаковывается, пакет передается на сетевой уровень и движется дальше.

Межсетевая маршрутизация

Маршрутизация на межсетевом уровне происходит примерно также как на сетевом, но с некоторыми дополнительными сложностями. Два уровням маршрутизации: внутреннему межшлюзовому протоклу и внешнему межшлюзовому протоколу. Поскольку каждая сеть в определенном смысле автономна, то для нее часто используют термин - ***автономная система*.**

Фрагментация
В каждой сети есть свой максимальный размер пакетов. Максимальный размер пакета колеблется от 48 байтов в ATM сети до 65 515 байтов в IP сети. Проблема возникает при попытке передать большой пакет через сеть, у которой максимальный размер пакета меньше.
Единственное решение - разрешить шлюзу разбивать пакет на фрагменты и отправлять каждый фрагмент независимо. В этом случае возникает проблема сборки фрагментов. Есть два подхода для этого.
 Первый делать фрагменты столь малыми, что любая сеть на их пути будет прозрачна для них.
Другой подход - разбив пакет на фрагменты трактовать каждый из них как обычный пакет. Сборка фрагментов происходит только в узле назначения.

Firewall
Компания может иметь сколь угодно сложную сеть, объединяющую много локальных сетей. Однако, весь трафик в сеть и из этой сети направляется только через один шлюз, где происходит проверка пакета на соответствие определенным требованиям. Если пакет не удовлетворяет этим требованиям, то он не допускает в или из сети. Шлюз приложений ориентирован на конкретные приложения. Межсетевые экраны(МЭ) разбиваются на группы по принципу насколько сильно используют информацию в пакете. В простейшем случае МЭ принимая решение сбросить или пропустить пакет смотрит только на адрес получателя и отправителя. В другом случае МЭ смотрит на протокол которому принадлежит пакет(в случае с установлением соединения было бы странно пропускать пакеты только в одном направлении, поэтому проверяется адрес только одного хоста). Самые сложные, они же и самые долгие МЭ анализируют содержание пакета(например текст для HTTP пакета)

Билет № 56.
Сетевой уровень в Интернет: адресация, протокол IPv4, протоколы ARP, RARP.

Каждая машина в Internet имеет уникальный IP адрес. Он состоит из адреса сети и адреса машины в этой сети. Все IP адреса имеют длину 32 разряда. Все адреса разделяются на классы. Всего есть пять классов адресов: A, B, C, D, E. Классы A позволяет адресовать до 126 сетей по 16 миллионов машин в каждой, B - 16382 сетей по 64K 000 машин, C - 2 миллиона сетей по 256 машин, D - групповая передача, E зарезервирован для развития. Адреса выделяет только NIC - Network Information Center.

Подсети
Все машины одной сети должны иметь одинаковый номер сети в адресе. По мере роста сети приходиться менять класс адреса. Перенос машины из одной сети в другую требует изменения маршрутизации. Решением этих проблем является разделение сети на части, которые извне видны как единое целое, но внутри каждая часть имеет свой адрес. Эти части называются подсети. Подсеть - это часть сети, не видимая извне. Изменение адреса подсети или введение новой подсети не требует обращения в NIC или изменений какой-либо глобальной базы данных.

Адресация.
Есть две таблицы. Первая показывает как достичь интересующей сети. Вторая - как достичь узел внутри сети. Когда поступает IP пакет, маршрутизатор ищет его адрес доставки в таблице маршрутизации. Если это адрес другой сети, то пакет передают дальше тому маршрутизатору, который отвечает за связь с этой сетью. Если это адрес в локальной сети, то маршрутизатор направляет пакет прямо по месту назначения. Если адреса нет в таблице, то маршрутизатор направляет пакет специально выделенному по умолчанию маршрутизатору, который должен разобраться с этим случаем с помощью более подробной таблицы. Такая организация алгоритма позволяет существенно сократить размер таблиц в маршрутизаторах. С появлением подсети структура адресов меняется. Теперь записи в таблице имеют форму (эта_сеть, подсеть, 0) и (эта_сеть, эта_подсеть, машина). Таким образом, маршруизатор подсети в данной локальной сети знает, как достичь любую подсеть в данной локальной сети, и как найти конкретную машину в своей подсети. Все что ему нужно – это знать ***маску подсети.*** С помощью логической операции И маршрутизатор выделяет адрес подсети с помощью маски. По своим таблицам он определяет как достичь нужной подсети или, если этого локальная подсеть данного маршрутизатора, как достичь конкретной машины.

Протоколы управления межсетевым взаимодействием

Internet Control Message Protocol.
Управление функционированием Internet происходит через маршрутизаторы с помощью протокола ICMP. Этот протокол выявляет и рассылает сообщения о десятках событий

Address Resolution Protocol – протокол определения адреса.
Ethernet адрес. Этот адрес имеет 48 разрядов. Сетевая карта знает только такие адреса и ничего об 32-разрядных IP. Как отобразить 32-разрядный IP адрес на адреса канального уровня, например, Ethernet адрес. Для отображения IP адреса на Ethernet адрес, в подсеть посылается запрос у кого такой IP адрес. Машина с указанным адресом шлет ответ. Протокол, который реализует рассылку запросов и сбор ответов - ARP протокол. Практически каждая машина в Internet имеет этот протокол. Для того чтобы узнать адрес в другой сети есть два решения - есть определенный маршрутизатор, который принимает все сообщения, адресованные определенной сети или группе адресов - proxy ARP. Этот маршрутизатор знает как найти адресуемую машину. Другое решение - выделенный маршрутизатор, который управляет маршрутизацией удаленного трафика. Машина определяет, что обращение идет в удаленную сеть и шлет сообщение на этот маршрутизатор.

Reverse Address Resolution Protocol (RARP) – обратный протокол определения адреса
Иногда возникает обратная проблема - известен Ethernet адрес, какой IP адрес ему соответствует. Он посылает запрос к RARP серверу: Мой Ethernet адрес такой то, кто знает соответствующий IP адрес? RARP сервер отлавливает такие запросы и шлет ответ. У этого протокола есть один существенный недостаток – пакеты с одним и тем же запросов рассылаются всем, увеличивает накладные расходы. Для устранения этого недостатка был предложен протокол BOOTP. В отличии от RARP, BOOTP использует UDP сообщения, которые рассылаются только маршрутизаторам

Билет № 57.
Сетевой уровень в Интернет: адресация, протокол IPv6.

Из – за того что распределение адресов ipv4 становится невозможным связи с ростом числа машин. Была поставлена задача придумать новый протокол такой, чтобы он удовлетворял требованиям:

1. Работа с миллиардами машин, даже при не эффективном распределении адресов;
2. Сократить размер таблиц маршрутизации;
3. Упростить протоколы, чтобы сделать маршрутизацию быстрее;
4. Обеспечить более высокую безопасность, чем существующий IP;
5. Обратить больше внимания на тип сервиса, особенно для приложений реального времени;
6. Расширить групповую адресацию, разрешив описание группы;
7. Разрешить роуминг для хоста без изменения его адреса;
8. Позволить эволюцию протоколов в будущем;
9. Разрешить совместное существование как старых таки новых протоколов.

В 1993 году был опубликован протокол SIPP - Simple Internet Protocol Plus, который был принят как IPv6. IPv6 не совместим с IPv4, но может работать с TCP, UDP, ICMP, IGMP,OSPF,BGP DNS

Первое и главное отличие IPv6 - более длинный адрес - 16 байт. Это решает одну из главных задач - неограниченное расширение Internet.
Второе - заголовок стал проще (всего 7 полей), что ускорило обработку и маршрутизацию.
Третье - он лучше поддерживает варианты в заголовке, что делает работу с ним более гибкой, позволяя опускать не нужные поля и вводить необходимые.
Четвертое - серьезно улучшена безопасность протокола. Идентификация и конфиденциальность - ключевые возможности нового IP.
Наконец, существенно улучшена работа с типом сервиса, особенно учитывая возрастающий multimedia трафик.

Существуют различные типы адресов IPv6: одноадресные (Unicast), групповые (Anycast) и многоадресные (Multicast).
Адреса типа Unicast хорошо всем известны. Пакет, посланный на такой адрес, достигает в точности интерфейса, который этому адресу соответствует.
Адреса типа Anycast синтаксически неотличимы от адресов Unicast, но они адресуют группу интерфейсов. Пакет, направленный такому адресу, попадёт в ближайший (согласно метрике маршрутизатора) интерфейс. Адреса Anycast могут использоваться только маршрутизаторами.
Адреса типа Multicast идентифицируют группу интерфейсов. Пакет, посланный на такой адрес, достигнет всех интерфейсов, привязанных к группе многоадресного вещания.

Билет № 58.
Протоколы внутренней маршрутизации (RIP, OSPF).

Динамическую маршрутизация. Один из наиболее популярных алгоритмов - **маршрутизация по вектору расстояния,** построен на идеях алгоритмов Беллмана-Форда и Форда-Фолкерсона. Изначально использовался в сети ARPA и используется по сей день под названием RIP алгоритма. В основе его лежит идея, что у каждого маршрутизатора в транспортной подсети есть таблица расстояний до каждого маршрутизатора в подсети. Периодически маршрутизатор обменивается такой информацией со своими соседями и обновляет информацию в своей таблице. Каждый элемент таблицы состоит из двух полей: первое - номер линии, по которой надо отправлять пакеты, чтобы достичь нужного места, второе - величина задержки до места назначения. Эта величина задержки может быть измерена в разных единицах: скачках, миллисекундах, длине очереди на линии и т.д. Каждые Т секунд маршрутизатор шлет своим соседям свой вектор задержек до всех маршрутизаторов в подсети. В свою очередь он получает такие же вектора от своих соседей. Кроме этого, он постоянно замеряет задержки до своих соседей.

OSPF - внутренний протокол маршрутизации шлюзов
Каждая такая сеть использует внутри свои алгоритмы маршрутизации и управления и называется **Автономной системой.** Алгоритмы маршрутизации, применяемые внутри АС, называются ***внутренними протоколами шлюзов***. Алгоритмы маршрутизации, применяемые для маршрутизации между АС, называются ***внешними протоколами шлюзов.*** Изначально в качестве внутреннего протокола шлюзов использовался протокол по вектору расстояния (RIP). Однако, по мере роста АС он начинал работать все хуже и хуже. Проблемы “счетчика до бесконечности”, медленная сходимость не получили удовлетворительного решения.
Требования к новому протоколу.
1. Алгоритм должен быть опубликован в открытой литературе - отсюда open.
2. Он не должен быть собственностью какой-либо компании.
3. Он должен уметь работать с разными метриками: расстоянием, пропускной способностью, задержкой и т.п. Он должен быть динамическим, т.е. реагировать на изменении в топологии сети автоматически и быстро.
4. Он должен поддерживать разные виды сервиса. Он должен поддерживать маршрутизацию для трафика в реальном времени одним способом, а для других другим. В IP пакете есть поле Type of service, которое не использовалось существующими в то время протоколами.
5. Он должен обеспечивать балансировку нагрузки и при необходимости разделять потоки по разным каналам. Все предыдущие использовали только один канал - наилучший.
6. Он должен поддерживать иерархию.
7. Должна быть усилена безопасность маршрутизаторов.

OSPF поддерживает три вида соединений и сетей:
1. Точка-точка между двумя маршрутизаторами;
2. Сети с множественным доступом и вещанием (большинство ЛВС);
3. Сети с множественным доступом без вещания (например, региональные сети с коммутацией пакетов).

OSPF абстрагируется от конкретных сетей, маршрутизаторов и хостов в форме ориентированного графа, каждая дуга в котором имеет вес, представляющий задержку, расстояние и т.п. Многие АС сами по себе большие сети. OSPF позволяет разбить их на **области,** где каждая область это либо сеть, либо последовательность сетей. Области не пересекаются. Есть маршрутизаторы, которые не принадлежат никакой области. Область - обобщение понятия подсети. Вне области ее топология не видна. Каждая АС имеет **остовную область**, называемую областью 0. Все области АС соединяются с остовной, возможно через тунелирование. Так что можно из одной области попасть в другую через остовную. Для того, чтобы поддерживать разные типы сервисов, OSPF использует несколько графов, один с разметкой как задержка, другой - пропускной способностью, третий - надежностью. Хотя все три требуют соответствующих вычислений, но зато получаем три маршрута, с оптимизированных по задержке, пропускной способности и надежности. Во время функционирования возникают три вида маршрутов: внутри области, между областями и между АС. Внутри области - вычислить этот маршрут просто - наикратчайший до маршрутизатора получателя внутри области. Маршрутизация между областями всегда выполняется в три этапа: от источника до остовной области, от остовной до области назначения, внутри области назначения. Этот алгоритм навязывает звездообразную топологию OSPF: остовная область – центр, ось, остальные области – лучи, спицы.

OSPF различает четыре класса маршрутизаторов:
1. Внутренний, целиком внутри одной области;
2. Пограничный, соединяющий несколько областей;
3. Остовной, принадлежащий остовной области;
4. АС пограничный, соединенный с маршрутизаторами других АС.

Когда маршрутизатор загружается он рассылает сообщение Hello всем своим соседям: на линиях точка-точка, группам маршрутизаторов в ЛВС с множественным доступом, чтобы получить информацию о своем окружении. В OSPF маршрутизаторы обмениваются данными не со своими соседями, а со смежными маршрутизаторами. . Этот выделенный маршрутизатор(смежные маршрутеризаторы) смежен всем другим. У выделенного маршрутизатора есть дублер, который имеет ту же информациою, что и основной. Периодически в ходе нормальной работы каждый маршрутизатор рассылает всем своим смежным маршрутизаторам сообщение LINK STATE UPDATE. В этом сообщении он передает информацию о состоянии своих линий и их стоимости в разных метриках для базы данных топологии соединений. Это сообщение в целях надежности идет с подтверждением. DATABASE DESCRIPTION – сообщение, содержащее состояние всех каналов в базе данных отправителя. Используя сообщение LINK STATE REQUEST маршрутизатор может в любой момент запросить информацию о любой линии у другого. Маршрутизаторы в остовной области делают все что было здесь описано, а так же они обмениваются информацией с пограничными маршрутизаторами, чтобы уметь вычислять наилучший маршрут от любого маршрутизатора остовной области до любого другого маршрутизатора.

Билет № 59.

Понятие автономной системы. Протокол внешней маршрутизации BGP.

Интернет на сетевом уровне можно рассматривать как объединение транспортных сред или сетей, которые называются автономными системами.

Автономная система – это сеть, охватывающая единую территорию, находящаяся под единым административным управлением и имеющая единую систему правил маршрутизации (политику маршрутизации) по отношению ко всем остальным сетям. В Интернете нет какой-либо регулярной специально предусмотренной структуры сетей. Это соединение большого числа сетей, среди которых можно выделить несколько магистральных (backbone). К этим магистральным сетям подключены региональные сети, к которым, в свою очередь, подключены локальные сети организаций. Все автономные системы взаимодействуют через IP-протокол. В отличие от других протоколов сетевого уровня IP-протокол с самого начала создавался для объединения сетей. Его целью было наилучшим образом передавать дейтаграммы от одной машины к другой, где бы эти машины ни находились.

Для маршрутизации между АС используется BGP - протокол пограничных шлюзов. Его предшественником был протокол EGP. Основное отличие BGP от OSPF проистекает из различия в целях. При маршрутизации внутри АС основная цель - наикратчайший маршрут. При маршрутизации между АС надо учитывать и ряд политических условий, вызванных политикой конкретной АС.
Типичными примерами таких ограничений могут быть: Трафик не должен проходить через определенные АС.
Такие правила вручную вводятся в каждый BGP маршрутизатор.
По степени интереса направления трафика через сеть, сети делятся на три категории.
Первая - **тупиковые сети** они никуда не ведут. У них только одна точка соединения с BGP графом. Они не могут использоваться для транзита.
Сети с **множественными соединениями**. Они могут использоваться для транзита если допускают его.
Транзитные сети - такие как остовные, которые предназначены для транзита трафика, возможно с некоторыми ограничениями.
Пара BGP маршрутизаторов взаимодействуют через TCP соединение. BGP - это протокол на основе вектора расстояний. Однако, вместо стоимости для каждого места в сети, он хранит конкретный маршрут. BGP протокол легко решает проблему “счета до бесконечности”.

Билет № 60.
Архитектура MPLS: Принципы организации и функционирования.

MPLS (**MultiProtocol Label Switching**) — это технология быстрой коммутации пакетов в многопротокольных сетях, основанная на использовании меток. **MPLS** разрабатывается и позиционируется как способ построения высокоскоростных IP-магистралей, однако область ее применения не ограничивается протоколом IP, а распространяется на трафик любого маршрутизируемого сетевого протокола.

Принцип коммутации

В основе **MPLS** лежит принцип обмена меток. Любой передаваемый пакет ассоциируется с тем или иным классом сетевого уровня (Forwarding Equivalence Class, FEC), каждый из которых идентифицируется определенной меткой. Значение метки уникально лишь для участка пути между соседними узлами сети **MPLS**, которые называются также маршрутизаторами, коммутирующими по меткам (Label Switching Router, LSR). Метка передается в составе любого пакета, причем способ ее привязки к пакету зависит от используемой технологии канального уровня. Распределение меток между LSR приводит к установлению внутри домена MPLS путей с коммутацией по меткам (Label Switching Path, LSP). Каждый маршрутизатор LSR содержит таблицу, которая ставит в соответствие паре «входной интерфейс, входная метка» тройку «префикс адреса получателя, выходной интерфейс, выходная метка». Получая пакет, LSR по номеру интерфейса, на который пришел пакет, и по значению привязанной к пакету метки определяет для него выходной интерфейс. Вся операция требует лишь одноразовой идентификации значений полей в одной строке таблицы. Это занимает гораздо меньше времени, чем сравнение IP-адреса отправителя с наиболее длинным адресным префиксом в таблице маршрутизации, которое используется при традиционной маршрутизации. Сеть **MPLS** делится на две функционально различные области — ядро и граничную область. Маршрутизаторы ядра занимаются только коммутацией. Все функции классификации пакетов по различным FEC, а также реализацию таких дополнительных сервисов, как фильтрация, явная маршрутизация, выравнивание нагрузки и управление трафиком, берут на себя граничные LSR. Таким образом, главная особенность **MPLS** — отделение процесса коммутации пакета от анализа IP-адресов в его заголовке.

Элементы архитектуры

Метка — это короткий идентификатор фиксированной длины, который определяет класс FEC. По значению метки пакета определяется его принадлежность к определенному классу на каждом из участков коммутируемого маршрута.

Стек меток

В рамках архитектуры **MPLS** вместе с пакетом разрешено передавать не одну метку, а целый их стек. Результат коммутации задает лишь верхняя метка стека, нижние же передаются прозрачно до операции изъятия верхней. Такой подход позволяет создавать иерархию потоков в сети **MPLS** и организовывать туннельные передачи.

Привязка и распределение меток

Под привязкой понимают соответствие между определенным классом FEC и значением метки для данного сегмента LSP. Привязку всегда осуществляет «нижний» маршрутизатор LSR, поэтому и информация о ней распространяется только в направлении от нижнего LSR к верхнему. Вместе с этими сведениями могут передаваться атрибуты привязки. Существуют два режима распределения меток: независимый и упорядоченный. Первый предусматривает возможность уведомления верхнего узла о привязке до того, как конкретный LSR получит информацию о привязке для данного класса от своего нижнего соседа. Второй режим разрешает высылать подобное уведомление только после получения таких сведений от нижнего LSR, за исключением случая, когда маршрутизатор LSR является выходным для этого FEC.

Построение коммутируемого маршрута

Сначала посредством многоадресной рассылки сообщений UDP коммутирующие маршрутизаторы определяют свое «соседство». После того как соседство установлено, LDP открывает транспортное соединение между участниками сеанса поверх TCP. По этому соединению передаются запросы на установку привязки и сама информация о привязке. Кроме того, участники сеанса периодически проверяют работоспособность друг друга, отправляя тестовые сообщения (keepalive message).

Заполнение таблиц меток по протоколу LDP

На стадии A каждое из устройств сети MPLS строит базу топологической информации, задействуя любой из современных протоколов маршрутизации. На стадии B маршрутизаторы LSR применяют процедуру нахождения соседних устройств и устанавливают с ними сеансы LDP. Далее маршрутизатор находит те сети для которых он является выходным и присваивает им случайную метку. На стадии D устройство LSR 1, которому известно значение метки для потока, адресованного в эту сеть, присваивает собственное значение метки данному FEC и уведомляет верхнего соседа (LSR 0) об этой привязке. Теперь LSR 0 записывает полученную информацию в свою таблицу. После завершения данного процесса все готово для передачи.

Билет № 61.
Транспортный уровень: сервис, примитивы, адресация, установление соединения, разрыв соединения, управление потоком и буферизация, мультиплексирование, восстановление разрывов.

Транспортный протокол - обеспечивает надежную передачу данных от одного абонента в сети другому.

Сервис для верхних уровней

Основная цель транспортного уровня - обеспечить эффективный, надежный и дешевый сервис для пользователей на прикладном уровне. Достижение этой цели обеспечивает сервис, предоставляемый сетевым уровнем. То, что выполняет работу транспортного уровня, называется **транспортным агентом**. Подобно сетевому уровню, транспортный уровень так же может поддерживать два вида сервиса, ориентированный на соединения и без соединений. Если транспортному уровню придет сообщение, что соединение на сетевом уровне неожиданно было разорвано, то он может установить новое сетевое соединение, с помощью которого выяснить, что произошло, какие данные были переданы, а какие нет и т.п. Идея транспортного уровня в том, чтобы сделать сервис транспортного уровня более надежным, чем сетевого. Другое важное соображение в том, что прикладная программа, опираясь на транспортный сервис, становится независимой от сети и может работать в сети с любым сервисом.

Качество сервиса

Транспортный уровень позволяет пользователю определить желаемые, допустимые и минимальные значения для различных параметров, характеризующих качество сервиса, в момент установки соединения. Далее транспортный уровень сам будет решать, сможет ли он с помощью сетевого сервиса удовлетворить запросы пользователя и до какой степени. Примеры параметров качества:

- Connection establishment delay - задержка на установку соединения определяет время между запросом на установку соединения и подтверждением о его установлении;
- Connection establishment failure probability - вероятность что соединение не будет установлено за время, равное задержке на установку соединения;
- Throughput - пропускная способность транспортного соединения определяет количество байт пользователя, передаваемых за секунду;
- Transit delay - задержка на передачу определяет время от момента, когда сообщение ушло с машины отправителя, до момента, когда оно получено машиной получателем;
- Residual error ration - доля ошибок при передаче. Теоретически этот параметр должен быть равен 0, если транспортный уровень надежно передает сообщение. На практике это не так;
- Protection - этот параметр позволяет определить уровень защиты передаваемых данных от не санкционированного доступа третьей стороной;
- Priority – приоритет позволяет пользователю указать степень важности для него разных соединений;
- Resilience – устойчивость определяет вероятность разрыва транспортным уровнем соединения в силу своих внутренних проблем или перегрузки.

Если требуемое качество недостижимо, то транспортный уровень сразу сообщает об этом пользователю, даже не обращаясь к получателю сообщения.

Примитивы транспортного уровня

Сервер приложения выполняет примитив LISTEN, в результате чего он блокируется до поступления запросов от клиентов. Клиент для установления соединения выполняет примитив CONNECT. Транспортный агент на стороне клиента блокирует клиента и посылает пакет с запросом на установление соединения серверу.
По примитиву CONNECT транспортный агент на стороне клиента шлет CONNECTION REQUEST TPDU.
Транспортный агент сервера, видя, что сервер заблокирован по LISTEN, разблокирует сервер и посылает CONNECTION ACCEPTED TPDU. После этого транспортное соединение считается установленным и начинается обмен данными с помощью примитивов SENT и RECEIVE.
По окончании обмена соединение должно быть разорвано. Есть два варианта разрыва соединения: симметричный и асимметричный. Асимметричный разрыв предполагает, что для разрыва соединения одна из сторон посылает DISCONNECT TPDU. Получив этот TPDU, другая сторона считает соединение разорванным. При симметричном разрыве каждое направление закрывается отдельно. Когда одна сторона посылает DISCONNECT TPDU, это значит, что с ее сторона больше данных не будет. На рис.6-5 показана диаграмма состояний при установлении и разрыве соединения.
Другой набор примитивов, так называемые, сокеты Беркли. Примитив SOCET создает новую точку подключения, резервирует для нее место в таблице транспортного агента. По примитиву BIND сервер выделяет сокету адрес. LISTEN не блокирующий примитив. Он выделяет ресурсы и создает очередь, если несколько клиентов будут обращаться за соединением в одно и то же время. Примитив ACCEPT блокирующий в ожидании запроса на соединение.

Когда клиент выполняет примитив CONNECT, он блокируется своим транспортным агентом и запускается процесс установления соединения. Когда он закончится, клиента разблокируют и начинается обмен данными с помощью примитивов SEND и RECEIVE. Разрыв соединения здесь симметричен, т.е. соединение считается разорванным если обе стороны выполнили примитив CLOSE.

Элементы транспортного протокола

Транспортный протокол должен решать следующие проблемы:

1. Адресация: как адресовать прикладной процесс, с которым надо установить соединение?
2. Как корректно установить соединение? Ведь пакеты могут теряться. Как отличить пакеты нового соединения от повторных пакетов, оставшихся от старого?
3. Как корректно разрывать соединение?

Адресация

Проблема адресации состоит в том, как указать с каким удаленным прикладным процессом надо установить соединение? Для этого используется TSAP - Transport Service Access Point. Аналогичное понятие существует и на сетевом уровне - IP адрес - SAP для сетевого уровня.

Это решение хорошо работает для часто используемого сервиса с длительным периодом активности, но в случае с временными приложениями нужно как-то запустить их и узнать их TSAP. Для этого используется протокол установления начального соединения. На каждой машине есть специальный сервер процессов, который как бы представляет все процессы, исполняемые на этой машине. Этот сервер слушает несколько TSAP, куда могут поступить запросы на TCP соединение. Если нет свободного сервера, способного выполнить запрос, то соединение устанавливается с сервером процессов, который переключит соединение на нужный сервер, как только он освободится.

Другой пример использовать сервер имён. Пользователь устанавливает соединение с сервером имен, для которого TSAP известен, и передает ему имя сервиса. В ответ сервер имен шлет надлежащий TSAP. Клиент разрывает соединение с сервером имен и устанавливает его по полученному адресу.

Установление соединения

Пакеты могут теряться, храниться и дублироваться на сетевом уровне. И из-за задержек приходить позже. Одно из возможных решений - временное TSAP. После того, как оно использовано, TSAP с таким адресом более не возникает. Другое решение - каждому транспортному соединению сопоставлять уникальный номер. Когда соединение разрывается, этот номер заносится в специальный список. Другой подход - ограничить время жизни пакетов.

На практике нам надо обеспечить, чтобы умерли не только сами пакеты, но и уведомления о них. Безопасный способ установления соединения. Все машины в сети оснащены таймерами. Каждый таймер двоичный счетчик достаточно большой разрядности, равной или превосходящей разрядность последовательных чисел, используемых для нумерации пакетов. При установлении соединения несколько младших разрядов этого таймера берут в качестве начального номера пакета. Главное чтобы последовательности номеров пакетов одного соединения не приводили к переполнению счетчика и его обнулению. Эти номера можно также использовать для управления потоком, в протоколе скользящего окна. Проблема возникает, когда машина восстанавливается после сбоя. Транспортный агент не знает в этот момент, какое число можно использовать для очередного номера Для того, чтобы избежать повторного использования порядкового номера, который уже был сгенерирован перед сбоем машины, вводится специальная величина по времени, которая образует запрещенную область последовательных номеров Проблема номеров может возникать по двум причинам. Либо потому, что машина генерирует слишком быстро пакеты и соединения, либо потому, что делает это слишком медленно. Чем больше разрядность счетчика последовательных номеров, тем дальше отодвигается момент попадания в запретную область. Троекратное рукопожатие предполагает, что машина 1 шлет запрос на установление соединения под номером x. Машина 2 шлет подтверждение на запрос x, но со своим номером *y*. Машина 1 подтверждает получение подтверждения с номером *y*.

Разрыв соединения

Разрыв соединения, как уже было сказано, может быть асимметричным или симметричным. Асимметричный разрыв может привести к потере данных. Симметричный разрыв - каждая сторона проводит самостоятельно, когда она передала весь имеющийся объем данных.

Управление потоком и буферизация

На транспортном уровне отправитель сохраняет все пакеты на случай, если какой-то из них придется посылать вторично. Если получатель знает об этом, то он может иметь лишь один пул буферов для всех соединений и, если пришел пакет, и ему нет буфера в пуле, то он сбрасывается, в противном случае сохраняется и подтверждается.

Если сетевой уровень не надежный, то на транспортном уровне отправитель вынужден сохранять все отправленные пакеты до тех пор, пока они не будут подтверждены. При надежном сетевом сервисе, наоборот

отправителю нет нужды сохранять отправленные пакеты, если он уверен, что у получателя всегда есть буфер для сохранения полученного TPDU. Если такой уверенности нет, то ему придется сохранять пакеты. Однако, и в первом и во втором случае возникает проблема размера буфера. При фиксированной длине буфера, естественно организовывать пул буферов одного размера. Однако при переменной длине пакетов проблема становится много сложнее. Если размер буфера устанавливать по максимальной длине пакета, то мы столкнемся с проблемой фрагментации. Если - по минимальной, то один пакет придется пересылать как несколько с дополнительными накладными расходами. Можно установить схему динамического согласования размера буфера при установлении соединения. Оптимальное соотношение между буферизацией на стороне отправителя или на стороне получателя зависит от типа трафика. Для низкоскоростного, нерегулярного трафика буферизацию лучше делать на обоих концах. В общем случае лучше всего решать вопрос о количестве буферов динамически. Здесь надо только позаботиться о решении проблемы потери управляющих пакетов.

Другую проблему представляет согласование доступного числа буферов и пропускная способность сетевого уровня. Эту проблему лучше всего решать динамически с помощью управляющих сообщений. Механизм управления потоком должен, прежде всего, учитывать пропускную способность подсети, а уже потом возможности буферизации. Располагаться этот механизм должен на стороне отправителя, чтобы предотвращать накопление большого числа не подтвержденных сообщений.

Мультиплексирование

В целях удешевления стоимости транспортных соединений можно отобразить несколько транспортных соединений на одно сетевое. Такое отображение называется нисходящим мультиплексированием.

В некоторых случаях наоборот, в целях увеличения пропускной способности по отдельным транспортным соединениям, можно отобразить транспортное соединение на несколько сетевых и по каждому сетевому иметь свое скользящее окно. Такое мультиплексирование называется восходящим.

Восстановление после сбоев

Надо восстанавливать работоспособность машины, включая и транспортный уровень. Предположим, сервер упал и старается восстановить функционирование. Прежде всего, ему надо узнать у клиента, какое TPDU было последним не подтвержденным и попросить перепослать его. В свою очередь клиент может находиться в одном из двух состояний: S_1 – есть не подтвержденное TPDU, либо S_0 – все TPDU подтверждены.

Если сервер упал, послав подтверждение, но до того, как он осуществил запись, то клиент будет находится после восстановления сервера в состоянии S_0, хотя подтвержденное TPDU потеряно. Пусть наоборот сервер сначала записал TPDU, а потом упал. Тогда сервер после сбоя найдет клиента в состоянии S_1 и решит, что надо перепослать не подтвержденное TPDU. В результате получим повторное TPDU.

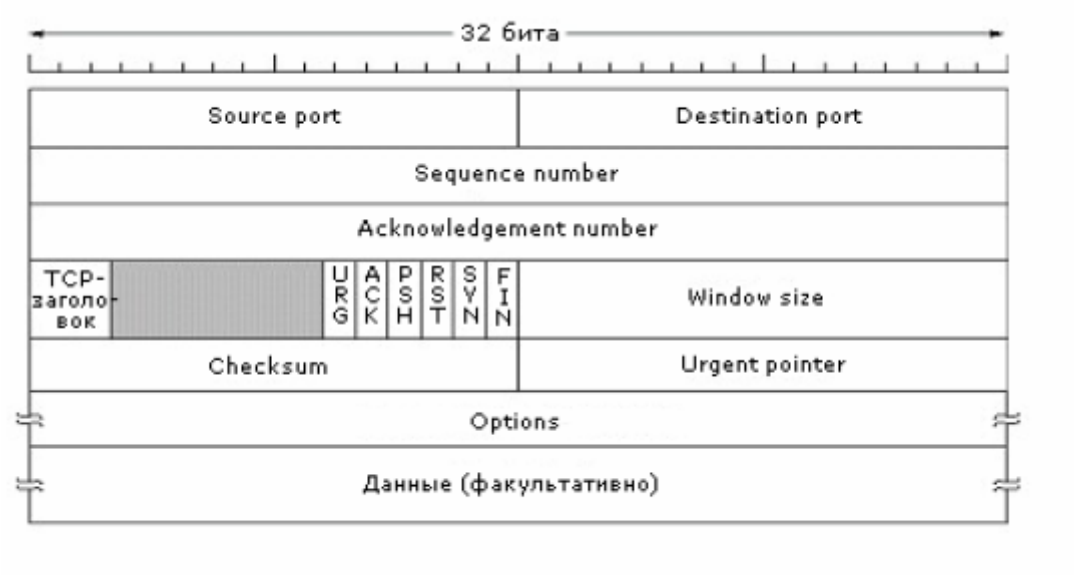
Надо, записав TPDU, информировать об этом приложение и только после этого слать подтверждение. При восстановлении надо опрашивать не только клиента на транспортном уровне, но и приложение.

Билет № 62.
Транспортный уровень в Internet (TCP, UDP). Сервис TCP, протокол, заголовок сегмента, управление соединениями, стратегия передачи, управление перегрузками, управление таймерами. Протокол UDP, беспроводной TCP и UDP. Способы ускорения обработки TPDU.

TCP - ориентированный на соединение и UDP - не ориентированный на соединение.
Поскольку сервис, реализуемый протоколом UDP - это практически сервис, реализуемый протоколом IP, с добавлением небольшого заголовка.
TCP (Transmission Control Protocol) - специально созданный протокол для надежной передачи потока байтов по соединению «точка-точка» через ненадежную сеть. TCP получает поток данных от прикладного процесса, дробит их на сегменты не более чем по 65 Кбайт (на практике не более 1,5 Кбайт) и отправляет их как отдельные IP-пакеты.
Поскольку IP-уровень не гарантирует доставку каждого пакета, то в задачу TCP входит определение потерь и организация повторной передачи потерянного. Поскольку на сетевом уровне в Internet соединения не поддерживаются, то сегменты могут поступать к получателю в неправильном порядке и задача TCP - восстановить этот порядок.
Доступ к TCP-сервису происходит через сокет. Сокет состоит из IP-адреса хоста и 16-разрядного локального номера на хосте, называемого порт. Сокеты создаются как отправителем, так и получателем. Порт - это TSAP для TCP. Каждое соединение идентифицируется парой сокетов, между которыми оно установлено. Один и тот же сокет может быть использован для разных соединений.
Порты с номерами до 256 зарезервированы для стандартного сервиса.(FTP, SSH, etc)
Все TCP-соединения - дуплексные, т.е. передача идет независимо в оба направления. TCP-соединение поддерживает только соединение «точка-точка».
TCP обеспечивает поток байтов, а не поток сообщений.
После того, как приложение передало данные TCP агенту, эти данные могут быть отправлены сразу на сетевой уровень, а могут быть буферизованы, в заголовке TCP-пакета есть флаг PUSH. Если он установлен, то это говорит о том, что данные должны быть переданы немедленно. Если для данных установлен флаг URGENT в заголовке, то все данные после этого по данному соединению передаются сразу и не буферизуются. Когда срочные данные поступают к месту назначения, то получателю передают их немедленно.
Каждый байт в TCP соединении имеет 32-разрядный номер. TCP-агенты обмениваются сегментами данных. Каждый сегмент имеет заголовок от 20 байтов и более (по выбору) и тело переменной длины. длина сегмента не должна превышать максимальную длину IP-пакета - 64 Кбайт. Во-вторых, каждая сеть имеет максимальную единицу передачи - MTU (maximum transfer unit), и каждый сегмент должен помещаться в MTU.
Основным протоколом, который используется TCP-агентом, является протокол скользящего окна. Это значит, что каждый посланный сегмент должен быть подтвержден. Одновременно с отправлением сегмента взводится таймер. Подтверждение придет либо с очередными данными в обратном направлении, если они есть, либо без данных, но с подтверждением. Подтверждение будет иметь порядковый номер очередного ожидаемого получателем сегмента. Если таймер исчерпается прежде, чем придет подтверждение, то сегмент посылается повторно.
TCP-протокол достаточно сложен и должен решать следующие основные проблемы:

- восстанавливать порядок сегментов
- убирать дубликаты сегментов, в каком бы виде они не поступали
- определять разумную задержку для time_out для подтверждений в получении сегмента
- устанавливать и разрывать соединения надежно
- управлять потоком
- управлять перегрузками

Заголовок TCP



Поля Source port и Destination port указывают сокеты на стороне отправителя и получателя соответственно. Sequence number и Acknowledgement number содержат порядковый номер ожидаемого байта и следующего ожидаемого, а не последнего полученного байта.

- Бит Urg используется вместе с полем Urgent pointer, которое указывает на начало области срочных данных.
- ACK - 1, если поле Acknowledgement number используется, в противном случае – 0.
- PSH - 1, если отправитель просит транспортного агента на стороне получателя сразу передать эти данные приложению и не буферизовать их.
- RST – используется, чтобы переустановить соединение, которое по какой-либо причине стало некорректным. Получение пакета с таким флагом означает наличие проблемы, с которой надо разбираться.

Поле Window size используется алгоритмом управления окном.
Поле Options используется для установления возможностей, не предусмотренных стандартным заголовком. Например, здесь часто указывается максимальный размер поля данных, допустимый по данному соединению.
Таймер для последовательных номеров сегментов тактируется с частотой 4 мксек., максимальное время жизни пакета - 120 сек.
Перегрузки
Поэтому каждый отправитель поддерживает два окна - обычное окно отправителя и окно перегрузки. Каждое показывает количество байтов, которое отправитель может послать. Фактически отправляемое количество байтов - минимум из этих двух величин.
Сначала окно перегрузки полагают равным размеру максимального сегмента для данного соединения. Если сегмент успешно (без time_out) был передан, то окно перегрузки увеличивается вдвое. Это увеличение будет происходить до тех пор, пока либо не наступит time_out и произойдет возврат к предыдущему значению, либо размер окна перегрузки не достигнет размера окна получателя. Этот алгоритм называется slow start - медленный старт.
Другой параметр управления перегрузками в Internet – порог (threshold). Алгоритм медленного старта при возникновении перегрузки устанавливает этот параметр равным половине длины окна перегрузки, а окно перегрузки - равным размеру максимального сегмента. Окно перегрузки растет экспоненциально до тех пор, пока не сравняется с порогом, после чего оно растет линейно, пока не достигнет размера окна получателя. На этом рост прекращается до первой перегрузки.
В основе используемого в протоколе TCP алгоритма выбора времени таймаута, предложенного Якобсоном в 1988 году, лежит специальная переменная RTT для получения оптимального значения величины time_out (Round Trip Time), значение которой постоянно модифицируется. В этой переменной хранится наименьшее время подтверждения. При каждой передаче сегмента замеряется величина задержки подтверждения M. Если при очередной передаче подтверждение поступило прежде, чем наступил time_out, значение переменной RTT немного уменьшают, в противном случае - увеличивают по хитрой формуле.
таймер настойчивости. Он позволяет бороться со следующего типа тупиками. Когда получатель посылает сообщение с нулевым размером окна, отправитель останавливает передачу и ждет сообщения об изменении размера окна. Наконец, получатель послал это сообщение, а оно было потеряно. Все ждут. Чтобы избежать такой ситуации, используют таймер настойчивости. Если он исчерпан, то отправитель шлет сообщение получателю, напоминая ему о проблеме размера буфера.

Еще один важный таймер - таймер функционирования. Если по какой-либо причине по соединению долго не посылали сообщений, то надо проверить, функционирует ли оно. Когда этот таймер исчерпан, то соответствующая сторона шлет другой стороне запрос: «Жива ли ты?» Если ответа не поступает, то соединение считается разорванным.

UDP

Internet поддерживает также транспортный протокол без соединений - UDP (User Data Protocol). Протокол UDP (User Datagram Protocol) предназначен для обмена дейтаграммами между процессами компьютеров, входящих в единую сеть с коммутацией пакетов. В качестве протокола нижнего уровня UDP-протокол использует IP.

Протокол UDP предоставляет прикладным программам возможность отправлять сообщения другим приложениям, используя минимальное количество параметров протокола. Этот протокол не обеспечивает достоверность доставки пакетов, защиты от дублирования данных или от сбоев в передаче. За исключением параметров приложения - номеров портов отправителя и получателя пакета, UDP практически ничего не добавляет к IP-дейтаграмме

Заголовок UDP

Source Port (16 бит). Порт отправителя. Это поле может содержать номер порта, с которого был отправлен пакет, когда это имеет значение (например, когда отправитель ожидает ответа). Если это поле не используется, оно заполняется нулями.

Destination Port (16 бит). Порт назначения - это порт компьютера, на который пакет будет доставлен.

Length (16 бит). Поле длины. Длина (в байтах) этой дейтаграммы, включая заголовок и данные. (Минимальное значение этого поля равно 8).

Checksum (16 бит). Поле контрольной суммы. Контрольная сумма UDP-пакета представляет собой побитное дополнение 16-битной суммы 16-битных слов (аналогично TCP). В вычислении участвуют: данные пакета, заголовок UDP-пакета, псевдозаголовок (информация от IP-протокола), поля выравнивания по 16-битной границе (нулевые).

Беспроводная весчь-выбор таймаута соптимизякан для перегрузок, а не ошибок при передаче в канале, поэтому нужны другие алгоритмы.

Быстрая обработка TPDU

основным препятствием для быстрой работы сети является программное обеспечение стека протоколов.

надо уметь быстро различать этот нормальный случай(обработку в режиме Established) от остальных специальных, например, разрыва соединения.

На стороне отправителя прикладной процесс через программное прерывание передает TPDU транспортному агенту в ядре. Агент с помощью проверок определяет, во-первых, какой случай имеет место: нормальный – отправка TPDU или специальный – разрыв соединения, во-вторых, что оправляется регулярное TPDU, а не специальное, и в-третьих, что окно получателя имеет достаточный размер. Если все условия выполнены, то может быть запущен ускоренный процесс отправки.

Ускоренная отправка - храним шаблон заголовка, и меняем в нем только некоторые поля, типа счетчика кадров.

транспортный агент на стороне получателя должен найти запись о соединении для поступившего сегмента TPDU. Для TCP-протокола эта запись может храниться в хеш-таблице. Ключом к этой таблице может служить информация о портах отправителя и получателя и их IP-адресах. Другой подход к поиску записи о соединении предложил Кларк – использовать последнюю использованную. Как показала практика, эта эвристика работает хорошо.

Две другие области ускорения – управление буферизацией и таймерами. Основная идея ускорения при управлении буферизацией – избегать излишнего копирования. Управление таймерами состоит в том, что, хотя таймер устанавливается для каждого TPDU, срабатывает он лишь для немногих TPDU. Общая схема, оптимизирующая работу с таймерами, заключается в следующем. Записи о таймерах связываются в список. В очередном элементе списка указывают, сколько тактов от срабатывания предыдущего таймера должно пройти, чтобы сработал текущий. Поэтому, если есть три таймера, которые должны сработать в моменты 3, 10 и 12, то список будет выглядеть, как 3, 7, 2

соответственно. При такой организации достаточно корректировать при каждом такте не все таймеры, а только первую запись в списке.

Билет № 63.
Безопасность и способы защиты данных в сетях ЭВМ: методы шифрования. Обычное шифрование. Рассеивание и перемешивание. Два основных принципа шифрования. Алгоритмы с секретными ключами (Алгоритм DES). Алгоритмы с открытыми ключами.

Безопасность информации — это состояние информации, характеризующееся ее защищенностью от *внутренних* и *внешних угроз,* т. е. от нарушения конфиденциальности, целостности, доступности, а также от незаконного тиражирования, которые наносят материальный и моральный ущерб владельцу или пользователю этой информации.
Угроза безопасности информации — это потенциально возможное преднамеренное или непреднамеренное происшествие, которое может оказать нежелательное воздействие на саму систему в сети ЭВМ, а также привести к потере безопасности информации, хранящейся в ней.
Уязвимость сети ЭВМ — это некая характеристика сети, которая делает возможным возникновение угрозы безопасности информации.
Атака на сеть ЭВМ — это действие, предпринимаемое злоумышленником, заключающееся в поиске и использовании уязвимости сети.

Три группы проблем,
1. Секретность:
• конфиденциальность — только санкционированный доступ к информации (никто не может прочесть ваши письма без вашего ведома);
• целостность — только санкционированное изменение информации (никто без вашего разрешения не может изменить данные о вашем банковском счете).
2. Идентификация подлинности пользователей и документов:
3. Надежность управления или доступность ресурсов и сетевых услуг:
• несанкционированное использование ресурсов (если вы получите счет за телефонные переговоры, которые вы не вели, вам это вряд ли понравится);
• обеспечение доступности ресурсов для авторизованных пользователей. Прежде чем обсуждать проблемы обеспечения безопасности следует рассмотреть основные виды работы с информацией в сети: передачу, обработку, хранение и представление.

В каком месте стека протоколов должна располагаться защита информации в сети? Одного такого места не существует: защита возможна на каждом уровне сети. Например, на физическом уровне для контроля доступа к каналу можно поместить кабель в опечатанную трубу, заполненную газом под давлением. Тогда любая попытка просверлить эту трубу приведет к падению давления газа, срабатыванию датчика давления и включению сигнала тревоги. На канальном уровне данные могут быть зашифрованы на одной машине, а расшифрованы на другой, и об этом шифре верхние уровни могут ничего не знать – шифрование канала, часто применяется в сетях. На сетевом уровне распространенным решением является наличие брандмауера. На транспортном уровне проблему секретности данных при передаче решают шифрованием всех сегментов транспортного соединения и применением так называемых сеансовых шлюзов.

Обычное шифрование. Исходный текст, называемый также открытым текстом, обрабатывают по определенному алгоритму со специальным параметром (ключом). При этом сам алгоритм шифрования может быть хорошо известен, а менять требуется только ключи. В результате этой обработки получают шифр-текст, или криптограмму. Так как у Злоумышленника нет ключа, быстро прочесть сообщение он не может, для этого ему необходимо вскрыть шифр, т.е. узнать алгоритм и ключ, использованные при шифровании этого сообщения.
Как проверить устойчивость алгоритма шифрования к взлому? Для этого алгоритм публикуют. Публикуя алгоритм шифрования, его автор даром получает консультации многих исследователей в этой области.__ Если ни один из них в течение пяти лет не объявит, что он вскрыл алгоритм, то такой алгоритм можно считать вполне надежным.
Активный злоумышленник – не только копирует сообщение, но и отправляет свои сообщения, имитируя настоящего отправителя. Искусство создания шифра называют *криптографией*, а искусство его вскрытия — *криптоанализом*. Вместе эти дисциплины образуют *криптологию*.

Шифрование замещением состоит в том, что буква или группа букв замещается другой буквой или группой букв из того же самого либо из другого алфавита (моноалфавитное и полиалфавитное замещение). Шифр Юлия Цезаря – замена каждой буквы в слове третьей буквой, следующей за ней в алфавите: а => г, б => д, и

т.д. Это так называемое моноалфавитное замещение, где ключом является 33-буквенная строка, соответствующая алфавиту. Здесь возможны 33! ключа. Даже если на проверку одного ключа компьютер будет тратить 1 мкс, то на расшифровку уйдет около 1013 лет.
Алфавитов шифрограммы может быть несколько, и они могут изменяться по определенному правилу, зависящему от ключа.
Чтобы прочесть сообщение, необязательно тупо перебирать все возможные варианты ключей. Найти необходимый ключ быстрее можно, используя знание частотных характеристик 151 языка: частоты встречаемости отдельных букв, двухбуквенных буквосочетаний, трехбуквенных сочетаний и т.д. Для этого надо подсчитать частоту букв в шифр-тексте и попытаться сопоставить наиболее часто встречающимся буквы в шифре с буквами, наиболее часто встречающимися в языке. Затем найти устойчивые буквосочетания и т.д. Следовательно, здесь большое значение имеют дополнительные сведения о шифрограмме: на каком языке написано исходное сообщение, его длина, типичные приветствия в данном языке и т.д. Чем длиннее сообщение, тем представительнее будет выборка для его анализа по частоте встречаемости букв и буквосочетаний.

Шифрование перестановкой состоит в изменении порядка набора букв без изменения самих букв. Для примера рассмотрим метод шифрования по столбцам. Выбираем ключ — последовательность неповторяющихся символов, которые нумеруем в соответствии с их местом в алфавите. Шифруемый текст размещается по строкам. Длина строки — длина ключа. В результате получаем массив, где столбцы нумеруются в соответствии с номером символа в ключе. Каждому столбцу соответствует символ ключа, который имеет определенный номер. Упорядочим столбцы по возрастанию этих номеров: сначала выпишем все символы первого столбца, затем символы второго столбца и т.д.
Для раскрытия этого шифра криптоаналитик прежде всего должен убедиться, что имеет дело с шифрованием перестановкой, Для этого он должен подсчитать частоту встречаемости букв в шифре, и если она соответствует частотным характеристикам языка, то это означает, что это именно метод перестановки. Намек на порядок столбцов могут дать устойчивые буквосочетания, имеющиеся в языке.

Алгоритмы с секретными ключами
Если раньше алгоритм был простой, а вся сложность шифрования заключалась в ключе, то теперь, наоборот, стараются алгоритм делать как можно изощреннее, чтобы криптоаналитик, получив как угодно много зашифрованного текста, не смог из него ничего извлечь.
Все алгоритмы шифрования с позиции использования ключа подразделяются на *алгоритмы с секретным ключом* и *алгоритмы с открытым ключом*. Алгоритмы с секретным ключом используют один ключ и для шифрования, и для дешифрования, поэтому их часто называют симметричными алгоритмами шифрования. Этот ключ является строжайшим секретом, известным только тому, кто шифрует сообщение, и тому, кто это сообщение расшифровывает. Слабым местом этих шифров является этап установки общего секретного ключа.
DES. Одним из широко известных шифров с секретным ключом является *шифр DES* (Data Encryption Standard). Создан на базе разработки фирмы IBM, был принят как стандарт в области шифрования в январе 1977 г. правительством США. Алгоритм DES состоит из 19 этапов. На первом этапе исходный текст разбивается на блоки по 64 бит каждый, и над каждым блоком выполняется перестановка. Последний этап является инверсией первой перестановки. Предпоследний этап заключается в обмене местами 32 самых левых битов и 32 самых правых битов. Для этапов со 2-го по 17-й с помощью специального преобразования исходного 56-разрядного ключа строятся 16 частных ключей, которые используются для преобразования данных. У алгоритма DES имеется два недостатка. Во-первых, он представляет собой моноалфавитное замещение с 64-разрядным символом, а всегда при подаче одних и тех же 64 разрядов исходного текста на вход, те же самые 64 разряда получают на выходе. DES сохраняет структуру сообщения, т. е. одни и те же поля исходного текста попадут в одни и те же места шифр-текста, чем может воспользоваться злоумышленник. Зная структуру исходного сообщения и длину его полей, он просто переставит необходимые поля в шифр-тексте, чтобы несанкционировано изменить сообщение. Во-вторых, для начала шифрования надо иметь сразу весь 64-разрядный блок исходного текста, а это не совсем удобно, если речь идет об интерактивных приложениях. Кроме того, эти 64 разряда надо накапливать в открытом виде, что делает схему шифрования уязвимой. Тройное шифрование (EDE-схема).

AES. Шифруемые данные представляют в виде двухмерных байтовых массивов размером 4 х 4 байт. Все операции производятся над отдельными байтами массива, независимо над столбцами и строками. На каждой итерации алгоритма выполняются следующие преобразования массива:

1. Операция Sub Bytes, представляющая собой замену каждого байта массива данных в соответствии со специальной таблицей кодирования.
2. Операция Shift Rows, представляющая собой циклический сдвиг влево всех строк массива данных за исключением нулевой. Сдвиг *i-й* строки массива (для $i = 1, 2, 3$) производится на i байт.
3. Операция MixColumns, выполняющая умножение каждого столбца массива данных, который рассматривается как полином степени 2^8, на фиксированный полином $a(x) = 3x^3 + x2 + x + 2$. Умножение выполняется по модулю $x^4 + 1$.
4. Операция Add Round Key, преобразующая массив данных с расширенным ключом итерации (наложение ключа). Процедура получения расширенного ключа для каждой итерации такова: над *i-м* столбцом массива данных ($i = 0... 3$) побитово выполняется логическая операция XOR с расширенным ключом W_{4r+I}, где r — номер текущей итерации алгоритма, начиная с 1. Количество итераций R алгоритма зависит от длины ключа. Перед первой итерацией алгоритма AES выполняется предварительное наложение расширенного ключа $W_0...W_3$ на открытый текст первых четырех слов итерации с помощью операции AddRoundKey. Последняя итерация отличается от предыдущих тем, что в ней нет операции MixColumns.
В алгоритме AES используются ключи шифрования трех фиксированных размеров: 128, 192, и 256 бит. Цель процедуры расширения ключа состоит в формировании необходимого числа слов расширенного ключа для их использования в операции AddRoundKey. Дешифрация выполняется посредством применения обратных операций в обратной последовательности.

Алгоритмы с открытыми ключами. Пусть имеются алгоритмы шифрования E и дешифрования D которые удовлетворяют следующим требованиям:
• *$D(E\{P\})$ равно исходному тексту P;*
• чрезвычайно трудно получить D, зная E
• E нельзя вскрыть через анализ исходных текстов.
Алгоритм шифрования E и его ключ, так называемый открытый ключ, публикуют или помещают таким образом, чтобы каждый мог их получить. Алгоритм D также публикуют, чтобы подвергнуть его изучению и проверке на стойкость, а вот ключ к нему хранят в секрете. Это так называемый секретный, или закрытый, ключ.
Взаимодействие двух абонентов A и B происходит следующим образом. Пусть A хочет послать B текст P. Абонент A шифрует текст $E_B(P)$, зная алгоритм и открытый ключ абонента B для шифрования. Абонент B, получив текст $E_B(P)$ и использовав секретный ключ и алгоритм D_B, вычисляет $D_B(E_B(P)) = P$. Никто не прочтет текст P кроме абонентов A и B, так как по условию алгоритм E_B нераскрываем, а алгоритм D_B нельзя вывести из E_B.
Пример: алгоритм RSA. Общая схема этого алгоритма следующая:
1. Выберем два больших (больше 10^{100}) простых числа p и q.
2. Вычислим $n = p \times q$ и $z = (p - 1) (q - 1)$.
3. Выберем простое d, взаимно простое по отношению к z.
4. Найдем e, удовлетворяющее условию $(e \times d) \bmod z = 1$.
Разобьем исходный текст на блоки длиной p таким образом, чтобы каждый блок как число не превосходил n. Для этого выберем наибольшее $к$, при котором выполняется условие $p = 2^к < n$. Шифр сообщения p получим, вычислив шифрованный текст $C = p^e \pmod n$. Для расшифровки найдем $P = C^d \pmod n$.
Для шифрования требуются числа e, n., представляющие собой открытый ключ, а для дешифрования — числа d, n - закрытый ключ.
Безопасность, обеспечиваемая применением этого метода шифрования, основана на высокой вычислительной сложности операции разложения на простые множители больших чисел. Один из основных недостатков алгоритма RSA — медленная работа.

Билет № 64.
Безопасность и способы защиты данных в сетях ЭВМ: протоколы установления подлинности документов и пользователей (аутентификация на основе закрытого разделяемого ключа, установка разделяемого ключа, проверка подлинности через центр раздачи ключей, установление подлинности протоколом Kerberos, установление подлинности, используя шифрование с открытым ключом). Электронная подпись (подпись с секретным ключом, подпись на основе открытого ключа). Сокращение сообщения.

Протоколы установления подлинности — аутентификации — позволяют процессу убедиться, что он взаимодействует с тем, с кем должен, а не с тем, кто лишь представляется таковым.
Не путать: *Авторизация* – проверка прав на выполнение тех или иных операций.
Если, например, к серверу обратился процесс с запросом удалить файл x.dat и объявил себя процессом «Вася», то сервер должен убедиться, что перед ним действительно Вася (аутентификация) и что Вася имеет право делать то, что просит (авторизация). Общая схема всех протоколов аутентификации следующая: сторон *A* и сторона *B* начинают обмениваться сообщениями между собой или с центром раздачи ключей (ЦРК). При этом предполагается, что ЦРК — всегда надежный партнер, т.е. его нельзя фальсифицировать. Протокол аутентификации должен быть устроен таким образом, чтобы даже в случае если злоумышленник перехватит сообщения между *A* и *B,* то ни *A,* ни *B* не спутают друг друга со злоумышленником.

Аутентификация на основе закрытого разделяемого ключа
Протокол ответа по вызову: одна сторона посылает некоторое число (вызов), а другая сторона, получив это число, преобразует его по определенному алгоритму и отправляет обратно. Увидев результат преобразования и зная исходное число, инициатор может определить, правильно сделано преобразование или нет. Алгоритм преобразования является общим секретом взаимодействующих сторон.
Идея атаки состоит в том, чтобы «заставить» Петю дать некоторое число, после чего на третьем шаге подсунуть ему это же число как свой вызов. На этот вызов Петя, согласно протоколу, ответит преобразованным ключом. В результате злоумышленник получит и число, и ключ, что ему и надо, чтобы выдать себя за Машу.
Существует несколько общих правил построения протоколов аутентификации:
• инициатор передачи должен доказать, кто он есть, прежде чем вы пошлете ему какую-либо важную информацию;
• инициатор и отвечающий должны использовать разные ключ**чи;**
• инициатор и отвечающий должны использовать начальные вызовы из разных непересекающихся множеств.

Установка общего закрытого ключа
Как стороны могут установить общий закрытый ключ? Протокол Диффи—Хеллмана. Прежде всего, Маша и Петя должны договориться об использовании двух больших простых чисел *n* и *g,* удовлетворяющих определенным условиям. Причем эти числа могут быть общеизвестны. Затем Маша выбирает большое число, например *x,* и хранит его в секрете, а Петя выбирает число *y* и также держит его в секрете. Маша посылает Пете сообщение вида $(n, g, g^x \bmod n)$, а Петя — отвечает сообщением вида $(g^y \bmod n)$. Теперь Маша выполняет операцию $(g^y \bmod n)^x$, а Петя — операцию $(g^x \bmod n)^y$. Теперь они имеют общий ключ $(g^{xy} \bmod n)$. Злоумышленник следит за всем этим, и единственное, что мешает ему вычислить *x* и *y,* это то, что неизвестен алгоритм с приемлемой сложностью вычисления логарифма от модуля для простых чисел. Слабое место – «чужой в цепочке».

Проверка подлинности через центр раздачи ключей
Идея протокола проверки подлинности через ЦРК состоит в следующем. Маша выбирает ключ сессии K_S. Используя свой ключ K_A, Маша посылает в ЦРК запрос на соединение с Петей. ЦРК знает Петю и его ключ K_B. С помощью этого ключа ЦРК сообщает Пете ключ сессии K_S и информацию о том, кто хочет с ним соединиться.
Однако это решение уязвимо – *атака подменой*. Пусть злоумышленник как-то убедил Машу связаться с Петей и скопировал весь обмен сообщениями. Позже он может воспроизвести этот обмен за Машу и заставить Петю действовать так, как если бы с ним говорила Маша. Существует несколько решений:
1. Использование временных меток.
2. Использование разовых меток. Можно комбинировать использование разовых и временных меток.
3. Смена ключей для каждой новой транзакции, для чего необходимо иметь заранее согласованный список одноразовых ключей. Ключ повторно использован быть не может. Наиболее часто используемое решение.

Установление подлинности с использованием протокола «Цербер»

Протокол установления подлинности «Цербер» используется многими действующими системами. В протоколе «Цербер» используется предположение, что все часы в сети хорошо синхронизованы, а также предполагается использование кроме рабочей станции *A* еще трех серверов:

• сервера установления подлинности (СП), проверяющего пользователей на этапе входа в систему (login);
• сервера выдачи квитанций (СВБ), обеспечивающего идентификацию квитанции;
• сервера *B,* обеспечивающего выполнение работы, необходимой *A*.

Сервер подлинности аналогичен ЦВК и знает секретный пароль для каждого пользователя, а СВБ выдает квитанции, которые подтверждают подлинность заказчиков работ.

Сначала пользователь садится за рабочую станцию и посылает открыто свое имя СП, который отвечает ключом сессии и квитанцией вида $<K_S, K_{TGS}(A, K_s)>$. Все это зашифровано закрытым ключом *A*. Когда второе сообщение приходит на рабочую станцию, у *A* запрашивается пароль, чтобы по нему установить K_A для расшифровки второго сообщения. Причем пароль перезаписывается с временной меткой, чтобы предотвратить его захват злоумышленником. Выполнив операцию login, пользователь может сообщить станции, что ему требуется сервер *B*. Рабочая станция обращается к СВБ за квитанцией для использования сервера *B*. Ключевым элементом этого запроса является ключ $K_{TGS}(A, K_s)$, зашифрованный закрытым ключом СВБ. В ответ СВБ посылает ключ K_{AB} для работы *A и B*.

Теперь *A* может обращаться непосредственно к *B* с этим ключом. Это взаимодействие сопровождается временными метками, чтобы защититься от подмены. Если позднее *A* понадобится работать с сервером С, то *A* должна будет повторить третье сообщение, но указать в нем сервер С.

Поскольку сеть может быть очень большой, то нельзя требовать, чтобы все использовали один и тот же СП, т.е. сеть разбивают на области, в каждой из которых будут свои СП и СВБ, взаимодействующие между собой.

Установление подлинности с помощью шифрования с открытым ключом

Установить взаимную подлинность можно с помощью шифрования с открытым ключом. Пусть Маша и Петя уже знают открытые ключи друг друга и используют, чтобы установить подлинность друг друга, а затем применяют шифрование с секретным ключом, которое на несколько порядков быстрее.

Ривст и Шамир предложили протокол, защищенный от атаки типа «чужой в цепочке». Это так называемый протокол с внутренним замком, идея которого заключается в том, что после обмена открытыми ключами Маша и Петя передают свои сообщения в два этапа, например, сначала только четные биты, а затем нечетные. В этом случае злоумышленник, имея только четные биты сообщения Пети, не может расшифровать всего сообщения, а значит, не может послать сообщение Маши, зашифрованное ее открытым ключом. Не может он и просто переслать сообщение Пети, так как у Маши и у Пети разные открытые ключи.

Электронная цифровая подпись

Подлинность многих юридических, финансовых и прочих документов устанавливается наличием подписи уполномоченного лица. Имеются способы, позволяющие отличить фотокопии от подлинника. Подпись на документе — это факт, подтверждающий, что лицо, подписавшее документ, либо является автором документа, либо знакомо с документом.

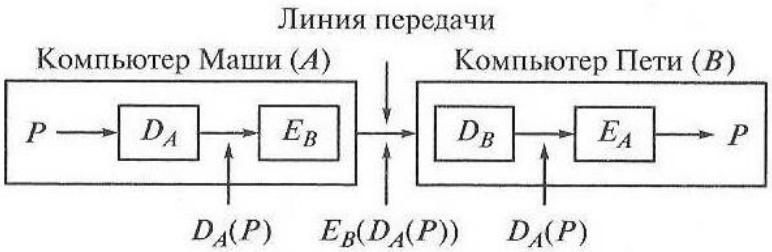
Проблема создания электронного аналога ручной подписи.

• получатель должен иметь возможность удостовериться в подлинности отправителя;
• отправитель не должен иметь возможность отречься от документа;
• получатель не должен иметь возможность подделать документ.

Подпись с секретным ключом

Одно из решений проблемы электронной подписи — наделить полномочиями третью сторону, которую знают все, которая знает всех и которой верят все. «Сердечный друг» (СД). Единственная слабость такого решения заключается в том, что злоумышленник может скопировать диалог между отправителем и получателем через СД и позже его повторить. Механизм временных меток позволяет ослабить эту проблему. Кроме того, сохранение последних ключей позволяет Пете заметить их повторное использование. Недостаток: необходимость доверять сердечному другу.

Подпись на основе открытого ключа



Предположим $E(D(P))=P$ дополнительно к $D(E(P))=P$ (этим свойством обладает алгоритм шифрования RSA). В этой схеме имеются два недостатка. Оба основаны на том, что схема работает до тех пор, пока Маша либо умышленно не рассекретит свой ключ, либо не изменит его в одностороннем порядке.

При наступлении судебного случая Петя предъявляет сообщение P и $D_A(P)$, а так как он не знает закрытый ключ Маши, то, значит, не мог подделать $D_A(P)$. При этом должно $E_A(D_A(P))=P$, в чем суд легко может убедиться.

Если Маша обращается в суд, т. е. предъявляет сообщение P и открытый ключ $E_A(P)$, это легко сопоставить с тем, что есть у Пети. Однако если Маша заявит, что у нее украли ключи, а сама тайно передаст их либо сменит, не сообщив об этом Пете, то в последнем случае текущий открытый ключ E_A будет неприменим к тому закрытому ключу $D_A(P)$, который предъявит Петя. При этом надо сопоставить даты передачи сообщения и смены ключей.

Профиль сообщения

Недостатки этих методов: подменяют задачу установления подлинности задачей шифрования, когда зачастую необходимо только установление подлинности; шифрование — медленная операция. Рассмотрим метод, который не требует шифрования всего сообщения. Он основан на использовании хэш-функции, которая по сообщению вычисляет битовую строку фиксированной длины. Эта функция, называемая профилем сообщения (Message Digest — MD), обладает тремя свойствами:

• у функции MD нет обратной функции;
• для заданного сообщения P вычислить функцию $MD(P)$ просто;
• имея MD(P), невозможно восстановить P\
• никто не сможет подобрать два таких сообщения, MD от которых будут одинаковыми.

Этот метод можно применять как с закрытым ключом, так и с открытым.

Билет № 65.
Служба DNS: основные функции, структуры данных, принципы функционирования.

Каждая машина в Интернете должна иметь IP-адрес. Однако оперировать числовыми IP-адресами неудобно, поэтому рассмотрим, как в Интернете можно использовать символьные имена вместо IP-адресов и как пользователь на абонентской машине может узнать IP-адреса других абонентских машин, зная их имена. Все Интернет-приложения позволяют при обращении к узлам сети вместо числовых адресов использовать имена, зафиксированные в специальной распределенной базе данных DNS, которая поддерживает иерархическую систему имен для идентификации абонентских машин или узлов в сети Интернет. Такой способ адресации на прикладном уровне называется символьной адресацией. Аналогия с почтовой службой. Проблемы: никакие компьютеры, включенные в сеть, не могут иметь одинаковых имен; преобразование имен в числовые адреса.
Когда Интернет был невелик, иметь дело с именами было довольно просто. Организация NIC создала регистратуру. Можно было послать запрос, и в ответ получить список имен и адресов - host file, этот файл регулярно рассылался всем машинам в сети. Все имена были простыми и уникальными. Компьютер просматривал файл и подставлял вместо имени реальный числовой адрес. По мере развития и расширения Интернета стало очевидно, что требуется распределенная оперативная система, работающая на новом принципе. Такая система была создана, и ее назвали доменной системой имен — DNS (Domain Name Service), а способ адресации в этой системе — способом адресации по доменному принципу. Также эту систему иногда называют региональной системой наименований.

Структура региональной системы имен
Доменная система имен — это метод, при котором в сетевой группе выделяется абонентская машина, отвечающая за назначение имен машинам в группе и обладающая полнотой информации о всех именах машин группы и их IP-адресах. При этом группы первого уровня могут быть объединены в группы второго уровня, группы второго уровня — в группы третьего уровня и т.д., причем ни одна группа не может входить в две и более групп. Каждая группа в такой иерархии называется доменом. Чтобы указать путь к интересующей нас машине, достаточно перечислить имена от самого верхнего домена до самого нижнего, содержащего интересующую нас машину. Домены в имени отделяют друг от друга точками. В имени может быть различное число доменов. Первым в имени стоит название абонентской машины — реального компьютера с IP-адресом. Это имя создано и поддерживается группой, к которой он относится. Группа входит в более крупное подразделение, которое, в свою очередь, является частью национальной сети.
Все пространство доменов распределено на зоны. Имена зон можно условно подразделить на организационные и географические. Организационные зоны: com, edu, gov, mil, net, org. В организационных зонах обычно размещаются непосредственно домены организаций. Каждая страна имеет свой географический домен из двух букв.
В доменном имени слева в конце цепочки доменных имен должно быть указано имя абонентской машины. Это имя может быть собственным или функциональным. Имена собственные каждый придумывает в меру своей фантазии. Имена функциональные вытекают из функций, выполняемых машиной: www — сервер HTTP (WWW); ftp — FTP-сервер; ns, nss, dns — сервер DNS (Name); mail — почтовый сервер; relay — почтовый сервер обмена; proxy — соответствующий proxy-сервер.
Доменная группа может создавать или изменять любые принадлежащие ей имена.

Поиск адреса по доменному имени
Доменная система работает автоматически, т. е. нам не надо разыскивать адрес, соответствующий имени, или подавать специальную команду для его поиска. Все компьютеры в Интернете могут пользоваться доменной системой.
Когда используют имя, например www.lvk.cs.msu.su, его необходимо преобразовать в IP-адрес. Для этого приложение формирует запрос к DNS-серверу, где работает DNS-служба. Эта служба — приложение, обладающее соответствующей базой данных, с помощью которой оно обслуживает такого рода запросы. Обработка имени DNS-сервером выполняется справа налево, т.е. сначала производится поиск адреса в самой верхней группе иерархии, а затем он постепенно опускается по иерархии, тем самым сужая область поиска. В целях сокращения поиска сначала опрашивается локальный узел DNS. При этом возможны три варианта ответов:
• местный сервер знает адрес, потому что этот адрес содержится в его базе данных.
• местный сервер знает адрес, потому что кто-то недавно уже запрашивал его, и он сохранил у себя в кэш-памяти этот адрес. Когда запрашивается адрес, DNS-сервер придерживает его у себя в кэш-памяти некоторое время на случай, если кому-нибудь потребуется тот же адрес, что повышает эффективность системы;

• местный сервер адрес не знает. В этом случае запускают ранее описанную процедуру опроса DNS-серверов доменов, указанных в имени справа налево.

Серверы имен

Нет и не может быть единого сервера, содержащего базу DNS, охватывающую весь Интернет (из-за вопросов безопасности, надёжности и производительности).

Чтобы сделать базу распределенной, все пространство имен доменов разбивают на непересекающиеся зоны. Границы каждой зоны определяет ее администратор. Каждая такая зона покрывает часть дерева доменов, и в нее входят серверы имен этих доменов. Обычно в каждой зоне есть основной сервер имен зоны и несколько вспомогательных серверов имен. Часто из соображений надежности сервер зоны располагают вне этой зоны.

Весь процесс поиска IP-адреса по имени домена реализуют серверы имен. Если запрос относится к юрисдикции того сервера имен, к которому обратились, т. е. запрашиваемый домен находится в ведении данного сервера имен, то этот сервер генерирует ответ, содержащий записи всех ресурсов, соответствующих запросу, и этот ответ считается авторитетным, т.е. содержащаяся в нем информация считается apriori верной.

Если запрос относится к удаленному домену, то сервер имен генерирует запрос к соответствующему удаленному серверу имен. Однако прежде чем обратиться к удаленному серверу имен обращающийся сервер посмотрит записи ресурсов в своей кэш-памяти. При этом записи в кэш-памяти не являются авторитетными. Время актуальности содержащейся в них информации определяет поле времени жизни.

Записи ресурсов

С каждым доменом связано множество ресурсов, отнесенных к этому домену. Записи об этих ресурсах хранятся в базе DNS. Когда происходит обращение к DNS-серверу с каким-либо именем, в ответ приходит не только IP-адрес, но и запись о ресурсах, соответствующих указанному имени.

Запись о каждом ресурсе состоит из пяти полей: «Имя домена» (Domain name), «Время жизни» (Time to live), «Класс» (Class), «Тип» (Type), «Источник полномочий» — SOA (Start Of Authority).

В поле «Имя домена» указывается имя домена, к которому относится эта запись. При обращении к базе DNS с таким ключом в ответ поступают все записи, у которых в этом поле указано заданное имя.

В поле «Время жизни» указывается интервал времени в секундах, в течение которого значение этого поля считается неизменным.

В поле «Класс» указывается значение IN, если ресурс, к которому относится эта запись, является ресурсом Интернета. Здесь могут быть указаны и другие значения, но они встречаются редко.

В поле «Источник полномочий» указывается имя источника информации о зоне сервера имен (об этом сервере будет сказано далее). Также здесь указывается адрес электронной почты администратора сервера имен и другая служебная информация. Если в этом поле указано значение A, то это означает, что в следующем поле указан IP-адрес этого ресурса. При указании в этом поле значения MX за ним следует имя машины, которая может получать почту для данного домена.

Записи типа NS указывают на серверы имен, относящиеся к домену верхнего уровня. Эта информация необходима при пересылке почты в другие домены.

Записи типов CNAME и PTR позволяют создавать псевдонимы. Например, человек может иметь несколько адресов электронной почты, но все они будут относиться к одному почтовому ящику.

Запись типа HINFO позволяет определять тип машины и операционной системы соответствующего ресурса.

Замечания по региональной системе имен

Доменная служба имен указывает на ответственного за поддержку имени, но ничего не сообщает о владельце компьютера, т. е. где эта машина находится географически (несмотря на коды стран).

Понятия доменного имени и сети не связаны. Часто доменные имена и сети перекрываются, и жестких связей между ними нет, т.е. две машины одного домена могут не принадлежать к одной сети. Например, системы io.cs.msu.su и fox.cs.msu.su могут находиться в совершенно разных сетях, поскольку доменные имена указывают ответственного за домен.

У машины может быть много имен. В частности, это верно для машин, предоставляющих какие-либо службы, которые в будущем могут быть помещены на другую машину. Когда эти службы будут перемещены, то имя, под которым такая машина выступала в качестве их сервера, будет передано новой машине-серверу вместе с услугами. При этом для внешних пользователей ничего не изменится, т.е. они будут продолжать пользоваться этой службой, запрашивая ее по имени, независимо от того, какой компьютер на самом деле реализует ее. Имена, по смыслу относящиеся к службе и называемые каноническими, в Интернете встречаются довольно часто.

Билет № 66.
Организация, функционирование и основные протоколы почтовой службы в Internet.

Поначалу возможности электронной почты сводились к передаче файлов с одним ограничением: первая строка файла должна была содержать адрес получателя. Со временем этого оказалось недостаточно в силу следующих обстоятельств:
• посылать одно и то же сообщение сразу нескольким получателям было неудобно;
• сообщение не имело внутренней структуры, что усложняло его обработку на машине;
• отправитель никогда не знал, получено сообщение или нет;
• невозможно перенаправить свои сообщения кому-то другому;
• интерфейс пользователя был неудобен, поскольку пользователь должен был от работы в редакторе файлов переходить в систему отправки файлов;
• было невозможно отправить в одном и том же сообщении и текст, и голос, и видео.

Архитектура почтовой системы включает в себя два основных компонента: агента пользователя (отвечает за интерфейс с пользователем, составление и отправку сообщений) и агента передачи сообщений (отвечает за доставку сообщения от отправителя к получателю).
Обычно почтовая система поддерживает пять базовых функций.
1. *Композиция.* Обеспечивает создание сообщений и ответов. Хотя для формирования тела сообщения может использоваться любой текстовый редактор, система автоматизирует заполнение многочисленных полей заголовка сообщения. Например, подстановка адреса при формировании ответа.
2. *Передача.* Обеспечивает передачу сообщения от отправителя к получателю без вмешательства пользователей.
3. *Отчет о доставке.*
4. *Визуализация сообщения.* Выполняет перекодировку сообщения, изменение формата и т.д.
5. *Размещение,* Определяет, что делать с сообщением: уничтожить после (до) прочтения или, если сохранить, то где. Поиск интересующего сообщения, перенаправление сообщения, повторное прочтение ранее полученного сообщения относятся также к данной функции.
Кроме указанных обязательных функций в большинстве почтовых систем имеется и ряд других функций. Например если пользователь уехал, он может перенаправить поступающие в его отсутствие сообщения куда-либо еще. Во многих системах пользователь может создавать так называемые внутренние почтовые ящики для поступающих сообщений; создавать лист рассылки, по которому одно и то же сообщение будет разослано всем его участникам; сортировать сообщения по определенным директориям в зависимости от их характеристик и многое другое.
Ключевой функцией всех современных почтовых систем является разделение почтового отправления на конверт сообщения и собственно сообщение. Система доставки использует только конверт, содержащий всю необходимую ей информацию о сообщении: адрес назначения, приоритет, секретность, требование об уведомлении и т.д. Сообщение внутри конверта имеет заголовок и тело. Заголовок содержит всю необходимую информацию о теле для агента пользователя, а тело предназначено исключительно для пользователя.

Агент пользователя
Агент пользователя — это обычно программа уровня приложений, способная выполнять определенный набор команд для получения, написания и композиции сообщения и ответа на сообщение, а также для работы с почтовым ящиком. При этом некоторые агенты используют командную строку, а некоторые — графический интерфейс.
Отправка почты. Чтобы послать сообщение, пользователь должен предоставить адрес назначения, само сообщение и другие его параметры, например приоритетность, секретность и т.п. Для создания сообщения может быть использован любой текстовый редактор, встроенный в агент пользователя. Все параметры должны быть заданы в формате, который понимает и с которым может работать агент пользователя. Большинство агентов пользователя ожидает адрес назначения в формате DNS; mailbox@location, где location — доменное имя, mailbox — ресурс в самом внутреннем (левом) домене в доменном имени. Агент пользователя также поддерживает лист рассылки, который позволяет рассылать одно и то же сообщение сразу нескольким пользователям. Причем сообщение размножается необязательно самим агентом, а там, где поддерживается лист рассылки.
Чтение почты. Прежде чем агент пользователя (далее АП) что-либо высветит на экране при загрузке, он просмотрит почтовый ящик на предмет новых поступлений и высветит на экране его содержимое с краткой

аннотацией каждого сообщения. В простых почтовых АП высвечиваемые поля встроены в АП, а в развитых — пользователь сам определяет, что показывать, а что нет (эта информация содержится в файле user profile).

Формат сообщений

Формат RFC 822. Сообщение состоит из простейшего конверта (описанного в RFC 821), полей заголовка, пустой строки и тела сообщения. Каждая строка заголовка — это строка ASCII-текста, содержащая название поля, двоеточие и какое-то значение. Стандарт 822 не различает четко заголовок и конверт. В современных почтовых системах это различие более четкое, и агент пользователя имеет дело с заголовком, а агент передачи — с конвертом, формируемым на основе заголовка.

Формат MIME (Multipurpose Internet Mail Extension). Когда Интернет только начинал развиваться, почтовые системы способны были передавать только текстовые сообщения на английском языке в формате ASCII. Для этих целей RFC 822 было достаточно. В наши дни этих возможностей уже недостаточно. Необходимо, чтобы почтовая система умела работать:

• с сообщениями на европейских языках (на французском, немецком и т.д.);
• с сообщениями не в латинском алфавите (на русском, арабском и т.д.);
• с сообщениями вне алфавита (на японском, китайском);
• с сообщениями, содержащими не только текст (звук, видео, графику).

Решение – MIME — многоцелевое расширение почтовой службы в Интернете. Основная идея — расширение RFC 822 в целях структурирования тела сообщения и введения правил кодировки ASCII-сообщений. Введение MIME повлияло на программы доставки и отправки сообщений. В формате MIME определены пять новых заголовков.

Заголовок MIME-Version сообщает агенту пользователя, что он имеет дело с MIME-сообщением и какая версия MIME используется.

Заголовки Content-Description и Content-Id характеризуют сообщение. Например, второй заголовок можно использовать для фильтрации сообщений.

Заголовок Content-Transfer-Encoding определяет подготовку сообщения для передачи через сеть, для чего используются четыре основных схемы. Простейшая схема применяется для передачи ASCII-текста — 7 бит на символ (для учета национальных алфавитов используется схема 8 бит на символ) при условии, что длина строки не превышает 1000 символов в строке. Для корректной передачи двоичных данных (например, исполняемого кода программ) используется схема base64 encoding, которая разбивает сообщение на блоки по 24 бит. Каждый блок разбивается на четыре группы по 6 бит каждая. Для сообщений, которые являются ≪почти≫ ASCII-сообщениями с небольшими исключениями, используется схема quoted-printable-encoding. Можно также указать и какую-то особую схему в поле Content-Transfer-Encoding.

В поле заголовка Content-Type указывается тип сообщения.

Передача сообщений

Основная задача системы передачи почтовых сообщений — надежная доставка сообщения от отправителя к получателю. Самым простым способом в этом случае является использование простого протокола передачи почты — SMTP (Simple Mail Transfer Protocol).

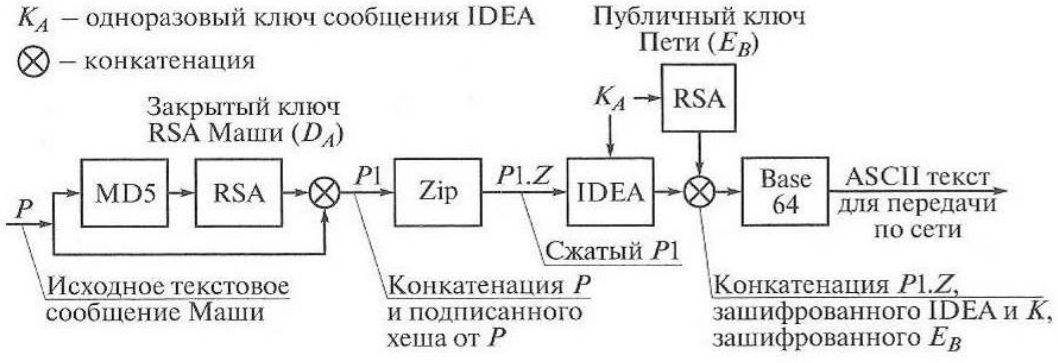
В Интернете почта передается следующим образом. Машина-отправитель устанавливает TCP-соединение с 25-м портом машины-получателя, на котором находится почтовый демон, работающий по протоколу SMTP. Этот порт принимает соединение и распределяет поступающие сообщения по почтовым ящикам машины-получателя. После установки соединения машина-отправитель работает как клиент, а машина-получатель — как сервер. Сервер посылает текстовую строку, идентифицирующую его и готовность принимать почту. Если он не готов принимать почту, то клиент разрывает соединение и повторяет всю процедуру позднее. Если сервер подтвердил свою готовность принимать сообщение, то клиент сообщает, от кого и кому оно предназначено. Если сервер подтвердил наличие получателя, то он дает команду клиенту и сообщение передается без контрольных сумм и подтверждений, так как TCP-соединение обеспечивает надежный поток байтов. Если сообщений несколько, то все они передаются. Обмен по соединению происходит в обоих направлениях.

Недостатки SMTP-протокола:

1. Длина сообщения не может превосходить 64 Кбайт.
2. Наличие time-out. Если время ожидания подтверждения у отправителя и получателя не согласовано, то один будет разрывать соединение, не дождавшись, тогда как другой просто будет очень загружен.
3. Возможность возникновения почтового урагана. Пусть машина-получатель имеет лист рассылки, где указана машина-отправитель, и наоборот. Тогда отправка сообщения по листу рассылки вызовет бесконечно долгие обмены сообщениями между этими машинами. Для преодоления этих проблем в RFC 1425 был

описан протокол ESMTP, по которому клиент сначала посылает команду EHLO, и если она отвергается сервером, это означает, что сервер работает по SMTP.
Почтовые шлюзы. Протокол SMTP хорош, когда обе машины находятся в Интернете. Однако это не всегда так. Многие компании в целях сетевой защиты соединяют свои сети через надлежащие средства либо используют другие протоколы. В этом случае отправитель передает сообщение шлюзу, тот его буферизует и позднее передает получателю. Проблемы:
1. Соответствие адресов.
2. Соответствие структур конвертов и заголовков.
3. Соответствие структуры тела сообщения.
Для простых неструктурированных ASCII-сообщений SMTP-шлюз способен решить такие проблемы.
Доставка получателю. Самый простой протокол для изъятия почты из удаленного почтового ящика — POP3 (Post Office Protocol), описанный в RFC 1225, позволяет входить в удаленную систему и выходить из нее, передавать письма и принимать их, а главное — он позволяет забирать почту с сервера и хранить ее на машине пользователя.
Более сложный протокол 1MAP — Interactive Mail Access Protocol, описанный в RFC 1064, позволяет одному и тому же пользователю заходить с разных машин на сервер, чтобы прочесть или отправить почту. Сервер в этом случае, по существу являющийся удаленным хранилищем писем, позволяет, например, получать доступ к письму не только по его номеру, но и по содержанию.
Часто используемый протокол — DMSP (Distributed Mail System Protocol, описанный в RFC 1056, не предполагает, что пользователь работает все время с одной и той же почтовой службой, т.е. пользователь может обратиться к серверу и забрать почту на свою локальную машину, после чего разорвать соединение. Обработав почту, он может ее отправить позднее, когда будет установлено очередное соединение.
Важными почтовыми сервисами являются:
• фильтры (спам-фильтры, сортировщик почты и т.д.) (программа procmail в UNIX-системах);
• возможность пересылки поступающей почты на другие адреса (настраивается через файл $HOME/.forward в UNIX);
• демон отсутствия (в UNIX-системах настраивается утилитой vacation);
• почтовый робот (программа, анализирующая входящие письма и отвечающая на них).

Конфиденциальность почты
PGP (Pretty Good Privacy). Почтовая служба с «вполне хорошей конфиденциальностью». Это полный пакет программ для обеспечения безопасности электронной почты, который включает в себя средства конфиденциальности, установления подлинности, электронной подписи и построения профиля сообщения в удобной для использования форме. Благодаря тому, что эта разработка качественная, работающая как на платформе UNIX, так и на платформах MS-DOS/Windows и Macintosh, и распространяющаяся бесплатно, она очень широко используется.



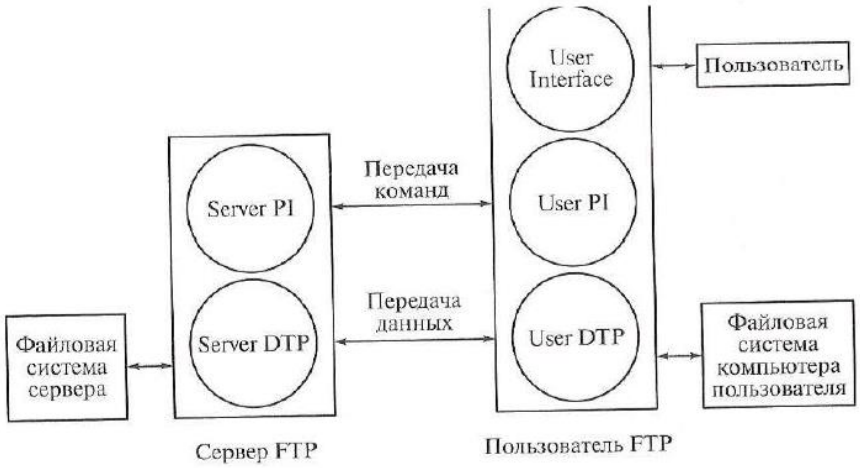
Секретный ключ для IDEA, строящийся автоматически в ходе работы PGP на стороне пользователя *A* и называемый ключом сессии — K_A, шифруется алгоритмом RSA с открытым ключом пользователя *B*. Также следует обратить внимание на то, что медленный алгоритм RSA используется для шифрования коротких фрагментов текста: 128-разрядного ключа для MD5 и 128-разрядного ключа для IDEA.
Служба PGP первых версий поддерживала три длины ключей:
• обычную — 314 бит (ключ может быть раскрыт за счет больших затрат);
• коммерческую — 512 бит (ключ может быть раскрыт специализированными организациями: ЦРУ, АНБ, ФСБ, МВД, ФБР);
• военную — 1024 бит (ключ, который в теории не может быть раскрыт пока никем на планете Земля).

В настоящее время существуют современные версии стандарта PGP, в которых используется алгоритм DSA, и размер ключа де-факто никак не ограничивается.
Формат PGP-сообщения включает в себя следующие поля:
• ID of E_B — ID ключа получателя;
• K_A — ключ сообщения, зашифрованный публичным ключом получателя;
• Sighdr — заголовок подписи;
• Time — метка времени подписи;
• ID of E_A — ID ключа отправителя;
• Type — тип алгоритма хэширования;
• MD5 hash — хэш (MD5) от сообщения, зашифрованный закрытым ключом отправителя;
• Msghdr — заголовок сообщения;
• File name — имя файла (генерируется при отправке);
• Message — исходное нешифрованное сообщение отправителя.

PEM (X.509). Почтовая служба с повышенной конфиденциальностью — имеет статус интернет-стандарта. Сообщения, пересылаемые с помощью PEM, сначала преобразуются в каноническую форму, в которой соблюдены соглашения относительно спецсимволов типа табуляции, последовательных пробелов и т. п. Затем сообщение обрабатывается алгоритмом MD5 или MD2, шифруется с помощью ключа DES (56-разрядный ключ) и передается с помощью кодировки base64. Передаваемый ключ защищается либо с помощью алгоритма RSA, либо с помощью ключа DES по схеме EDE. Управление ключами в PEM предполагает использование центров их сертификации. Каждый сертификат указывает уникальный порядковый номер и хеш MD5.

Билет № 67.
Служба FTP: организация, протокол.

Протокол передачи файлов (File Transfer Protocol — FTP) предназначен для решения следующих задач:
• разделение доступа к файлам на удаленных абонентских машинах;
• прямое или косвенное использование ресурсов удаленных компьютеров;
• обеспечение независимости клиента от файловых систем удаленных абонентских машин;
• эффективная и надежная передача данных.
FTP — это протокол прикладного уровня, который, как правило, использует в качестве транспортного протокола TCP. Его нельзя использовать для передачи конфиденциальных данных, поскольку он не обеспечивает защиты передаваемой информации и передает между сервером и клиентом открытый текст. Сервер FTP может потребовать от клиента FTP аутентификации (т.е. при установлении соединения с сервером FTP-пользователь должен будет ввести свой идентификатор и пароль). Однако и пароль, и идентификатор пользователя будут переданы от клиента на сервер открытым текстом.



• User Interface — пользовательский интерфейс работы с FTP;
• User PI (User Protocol Interpretator) — интерпретатор команд пользователя. Эта программа взаимодействует и с Server-PI, чтобы обмениваться командами управления передачей данных по каналу передачи команд, и с модулем User DTP, который осуществляет непосредственную передачу данных по каналу передачи данных;
• User DTP (User Data Transfer Process) — модуль, осуществляющий обмен данными между клиентом и сервером FTP по каналу передачи данных по командам от модуля User PI. Этот объект взаимодействует с файловой системой пользователя и объектом Server DTP;
• Server PI (Server Protocol Interpretator) — модуль управления обменом данных со стороны сервера по каналу передачи команд;
• Server DTP (Server Data Transfer Process) — модуль обмена данными со стороны сервера по каналу передачи данных;
• сервер FTP — собственно сервер FTP, который состоит из модуля Server PI управления передачей и модуля Server DTP, осуществляющего передачу;
• пользователь FTP — модуль клиента FTP, состоящий из-модуля управления передачей User PI и модуля, осуществляющего передачу, User DTP.

FTP поддерживает сразу два канала соединения: канал передачи команд (и статусов их обработки) и канал передачи данных. Канал передачи данных может использоваться для передачи и в одном, и в другом направлениях, кроме того, он может закрываться и открываться по командам управляющих модулей в процессе работы. Канал передачи команд открывается с установлением соединения и используется только для передачи команд и получения ответов после их обработки.

Алгоритм работы FTP следующий:
1. Сервер FTP устанавливает в качестве управляющего соединение с портом 21 TCP, который всегда находится в состоянии ожидания соединения со стороны FTP-клиента.
2. После установки управляющего соединения модуля User PI с модулем Server PI, клиент может отправлять на сервер команды. FTP-команды определяют параметры соединения передачи: роли участников соединения (активная или пассивная), порт соединения (как для User DTP, так и для Server DTP), тип передачи, тип передаваемых данных, структуру данных и управляющие директивы, обозначающие действия, которые пользователь хочет совершить, например сохранить, считать, добавить или удалить данные иди файл.

3. После согласования всех параметров работы канала передачи данных один из участников соединения, который является пассивным (например, клиентский модуль User DTP), переходит в режим ожидания открытия соединения на заданный для передачи данных порт. После этого активный модуль (например, Server DTP) открывает соединение и начинает передачу данных.
4. После окончания передачи данных соединение между Server DTP и User DTP закрывается, но управляющее соединение Server PI — User PI остается открытым. Пользователь, не закрывая сессии FTP, может еще раз открыть канал передачи данных, передать необходимую информацию и т.д.

Протокол FTP можно использовать при передаче файлов не только между клиентом и сервером, но и между двумя FTP-серверами. Для этого пользователь сначала устанавливает управляющие соединения с двумя FTP-серверами, а затем устанавливает между ними канал передачи данных. В этом случае управляющая информация передается через модуль User PI, а данные транслируются через канал Server 1 DTP - Server2 DTP.

Основу передачи данных в протоколе FTP составляют механизм установления соединения между соответствующими портами и механизм выбора параметров передачи. Каждый участник FTP-соединения должен поддерживать 21 порт передачи данных по умолчанию. По умолчанию User DTP использует тот же порт, что и для передачи команд (обозначим его U), а Server DTP использует управляющий порт с номером L1. Однако, как правило, участниками соединения используются порты передачи данных, выбранные для них User PI, поскольку из всех управляющих процессов, участвующих в соединении, только он может изменять порты передачи данных как у User DTP, так и у Server DTP.
Пассивная сторона соединения должна до подачи команды начала передачи слушать свой порт передачи данных. Активная сторона, подающая команду на начало передачи, определяет направление перемещения данных.
После установки соединения между Server DTP и User DTP начинается передача. Одновременно по каналу Server PI — User PI передается уведомление о получении данных. Протокол FTP требует, чтобы управляющее соединение было открыто все время пока по каналу обмена данными идет передача. Сессия FTP считается закрытой только после закрытия управляющего соединения.

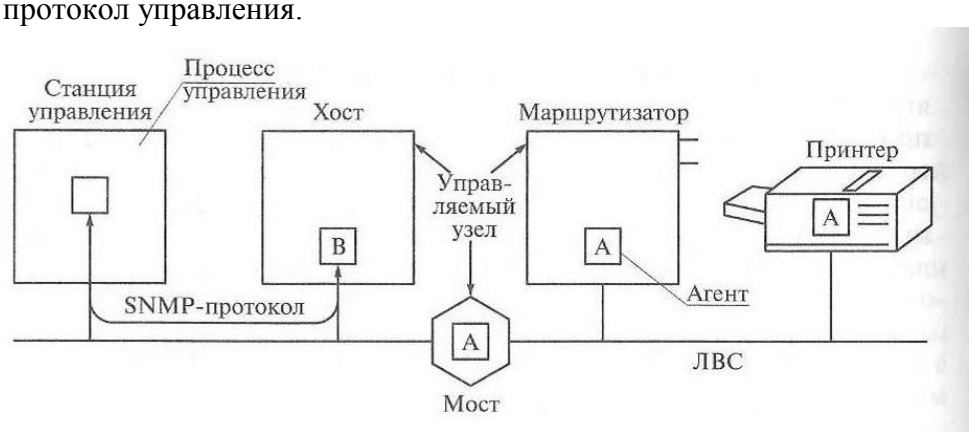
Как правило, сервер FTP ответственен за открытие и закрытие канала передачи данных. Сервер FTP должен самостоятельно закрывать канал передачи данных в следующих случаях:
• сервер закончил передачу данных в формате, требующем закрытия соединения;
• сервер получил от пользователя команду на прерывание соединения;
• пользователь изменил параметры порта передачи данных;
• было закрыто управляющее соединение;
• возникли ошибки, при которых невозможно возобновить передачу данных.

FTP-протокол имеет двух ≪братьев≫: TFTP (Trivial FTP) и SFTP (SSH FTP).
TFTP —это простейший протокол передачи файлов, работающий поверх транспортного протокола UDP и обеспечивающий выполнение только самых элементарных операций передачи файлов: запись и считывание файлов. SFTP (SSH File Transfer Protocol) —это FTP-протокол для передачи файлов через SSH-соединение. Также он предназначен для копирования и выполнения других операций с файлами поверх надежного и безопасного соединения по протоколу SSH. SFTP —это новый протокол, разработанный специально для надежной передачи файлов.
SSH (от *англ.* secure shell —безопасная оболочка) — это протокол удаленного терминального доступа к узлам сети, который позволяет регистрироваться на удаленном узле сети, выполнять на нем команды, а также копировать и перемещать файлы между компьютерами. Протокол SSH организует защищенное безопасное соединение поверх небезопасных каналов связи. В состав типичной реализации протокола SSH входит также и примитивный клиент для удаленного копирования файлов через SSH-канал —SCP (Secure CoPy). В сравнении с довольно примитивным протоколом SCP протокол SFTP позволяет выполнять намного больше операций с файлами, например, продолжать передачу файла после разрыва соединения или удалять файл на сервере. Для протокола SFTP имеются графические и псевдографические клиенты. При этом сам по себе протокол SFTP не обеспечивает безопасность работы, это делает нижерасположенный протокол SSH.

Билет № 68.
Служба управления сетью: организация, протокол SNMP, структура базы данных MIB.

Протокол управления сетью — SNMP
Когда сеть из компьютеров охватывает небольшие пространства, то в случае возникновения неполадок можно обойти все помещения и проверить работоспособность каждого устройства и его программного обеспечения. Когда сеть охватывает большие территории и включает в себя оборудование, принадлежащее разным организациям, то такой обход уже невозможен. Требуются адекватные средства для управления сетью. В 1990 г. была опубликована первая версия протокола управления сетью (Simple Network Management Protocol — SNMP v.l). В RFC 1155 и RFC 1157 было описано, как организовано систематическое наблюдение за сетью (какая, как и где может накапливаться информация) и управление ею (как и какие параметры работы устройств сети можно изменять). Этот протокол получил широкое распространение и был реализован практически во всех устройствах, используемых в сетях. Ряд недостатков SNMP v.l: например, недостаточно были проработаны вопросы безопасности. Во второй версии протокола SNMP (RFC 1441... 1452) была введена криптографическая защита механизма аутентификации. Далее рассматривается SNMP v.2.

Модель управления, принятая в протоколе SNMP, использует четыре типа сущностей:
• станции управления;
• управляемые устройства;
• управляющая информация;
• протокол управления.



Управляют сетью станции управления, т. е. компьютеры, на которых выполняются процессы, собирающие и накапливающие информацию об управляемых устройствах в сети. Сбор этой информации происходит по запросу от управляющей станции к управляемому устройству. Запросы, передача и другие действия выполняются с помощью команд протокола SNMP.
На управляемых устройствах работают специальные SNMP-агенты, которые выполняют команды, передаваемые с помощью SNMP-протокола, и фиксируют определенный набор параметров функционирования управляемого устройства. Управляемым устройством может быть маршрутизатор, мост, рабочая станция, устройство печати, т. е. любое устройство, где может работать SNMP-агент. Каждый агент поддерживает локальную базу данных MIB (Management Information Base). В этой базе хранится информация о состоянии агента, история его функционирования и переменные, характеризующие работу устройства, где функционирует агент.

Структура управляющей информации — SMI
В сети используется аппаратура сотен различных производителей. Естественно, агент должен формировать данные о функционировании управляемого устройства в некотором унифицированном виде, например, по составу или способу представления независимо от того, кто изготовил это устройство.
В соответствии с терминологией, принятой в стандарте протокола SNMP, переменную, в которой агент накапливает информацию, будем называть объектом. Все объекты собираются в группы, определяемые стандартом, а группы — в модули. Чтобы все объекты имели единые правила идентификации, поступают следующим образом. Строят дерево стандарта, в котором отражают иерархию используемых понятий, и это дерево является поддеревом дерева стандартов.
Коллекцию объектов, которыми можно управлять с помощью протокола SNMP, определяет база управляющей информации — MIB. Все объекты этой базы подразделяются на 10 групп, соответствующих 10 узлам, смежных узлу MIB-2 в дереве стандартов.
SMI (Structure of Management Information) — это в определенном смысле язык для определения структур данных, представляющих собой объекты в базе данных MIB.

Управление в сети с помощью протокола SNMP

SNMP-протокол определяет пять типов сообщений, которыми обмениваются станция управления и управляемое устройство:

• get-request — получить значение одной или нескольких переменных;

• get-next-request — получить значение одной или нескольких переменных, следующих после указанной переменной;

• set-request — установить значение одной или нескольких переменных;

• get-response — выдать значение одной или нескольких переменных. Это сообщение возвращается агентом станции управления в ответ на операторы get-request, get-next-request и setrequest;

• trap — уведомить станцию управления, когда что-либо произошло с агентом.

Первые три из этих сообщений использует станция управления, а последние два — управляемое устройство. Так как четыре из пяти SNMP-сообщений реализуются простой последовательностью типа «запрос—ответ», SNMP-протокол использует UDP-протокол. Это означает, что запрос от станции управления может не дойти до управляемого устройства, как и отклик от управляемого устройства — до станции управления. В этом случае будет задействован механизм time-out и выполнена повторная передача.

Станция управления отправляет все три запроса на UDP-порт 161. Управляемое устройство устанавливает ловушки (программные прерывания trap) на UDP-порт 162. Так как используются два разных порта, одна и та же система может выступать и как станция управления, и как управляемое устройство.

При взаимодействии между станцией управления и управляемым устройством используется пароль, представляющий собой 6-символьную строку, которую в SNMP v.l передавали в открытом виде. В операторах get, get-next и set станция управления устанавливает идентификатор запроса (request ID), который возвращается управляемым устройством в сообщении get-response, что повышает безопасность при взаимодействии. Это поле также позволяет станции управления выдать несколько запросов одному или нескольким устройствам, а затем отсортировать полученные отклики. Статус ошибки (error status) — это целое число, которое возвращается агентам и указывает на ошибку.

Билет № 69.
Веб-технологии: Протокол HTTP и его безопасная версия. Технологии на стороне сервера: CGI, модули для веб-сервера. Аутентификация и управление сеансами в HTTP.

Всемирная паутина WWW основывается на использовании гипертекста. Идея создать сеть из документов, расположенных на разных машинах и связанных гиперссылками, сформулирована Т. Бернерс-Ли в 1989 г. Начали использовать Web в январе 1992 г. в Женеве. Т. Бернерс-Ли предложил хранить документы на компьютерах, которые он назвал веб-серверами. Для реализации этой идеи были необходимы специальный протокол, умеющий работать с гипертекстовыми документами, средство описания документов и средство визуализации документа, собранного из отдельных документов, соединенных ссылками. Специальный протокол HTTP (Hyper Text Transmission Protocol) используется в Интернете с 1990 г. Для описания документов и связывания их гиперссылками служит язык HTML (Hyper Text Markup Language), а для просмотра документов используются специальные программы — браузеры. Всемирная паутина состоит из серверов, которые предоставляют доступ к хранящейся в них информации через графический интерфейс. Графический интерфейс реализуется через специальное программное обеспечение, работающее на абонентских машинах сети, — браузеры. Способность предоставлять информацию в виде видео, аудио, текста и изображений через стандартный набор элементов графического интерфейса, который не зависит от платформы, делает Web привлекательным ресурсом для всех категорий пользователей.

HTTP является текстовым протоколом, а это означает, что http-запросы можно послать web-серверу. Общий вид HTTP-запроса следующий:

Запрос = Метод SP URI-Запроса SP Версия-HTTP CRLF
Заголовок-Запроса CRLF
Заголовок-Запроса CRLF
Заголовок-Запроса CRLF
CRLF
[Тело запроса]

В HTTP-запросе могут использоваться следующие методы:
Метод = "GET" | "HEAD" | "PUT" | "POST" | " DELETE" | "LINK") "UNLINK" I дополнительный_метод
Метод GET служит для получения любой информации, идентифицированной URI-запросом, Если URI-запрос ссылается на процесс, выдающий данные, в качестве ответа будут выступать данные, сгенерированные указанным процессом (если они не являются выходными данными процесса), а не код самого процесса. Согласно стандарту HTTP многократное повторение одного и того же запроса GET должно приводить к одинаковым результатам (при условии, что сам ресурс не изменился за время между запросами), что позволяет кэшировать ответы на него.
Метод HEAD аналогичен методу GET за исключением того, что клиенту возвращается только заголовок сообщения-ответа (усеченный GET). Этот метод в основном используется для тестирования гиперссылок и проверки доступа к ресурсам.
Метод PUT служит для сохранения передаваемого на сервер ресурса с идентификатором URI.
Метод POST предназначен для передачи серверу информации, включенной в запрос как дополнительной к ресурсу, указанному в поле URI-запроса. Метод POST был разработан как общий для осуществления следующих целей:
• аннотация существующих ресурсов;
• добавление сообщений в группы новостей, почтовые списки и другие подобные группы статей;
• доставка блоков данных процессам, обрабатывающим данные;
• расширение баз данных через операцию добавления.
В отличие от метода GET при многократном повторении одного и того же запроса с методом POST можно получать разные результаты (например, после каждой отправки комментария в форум будет появляться новая копия этого комментария).
Метод DELETE используется для удаления ресурса, определенного идентификатором URI.
Как правило, методы PUT и DELETE в современных web-серверах запрещены, и управление данными осуществляется только через метод POST.

Web-серверы — обработчики HTTP-запросов
Протокол HTTP основан на парадигме запрос-ответ. Браузер устанавливает соединение с web-сервером и посылает ему запрос, содержащий метод запроса, URI и версию протокола, за которой следует сообщение в формате MIME, включающее в себя управляющую информацию запроса, информацию о клиенте и может быть тело сообщения. Для адресации на прикладном уровне большинство протоколов используют

универсальные идентификаторы ресурса — URI (Universal Resource Identifier). Самые известные примеры идентификатора URI — это URL и URN.
URL — это идентификатор URI, который помимо идентификации ресурса предоставляет еще и информацию о ее местонахождении. Изначально URL предназначался для обозначения мест расположения ресурсов (чаще всего файлов) во Всемирной паутине. Сейчас URL применяется для обозначения адресов почти всех ресурсов Интернета и позиционируется как часть более общей системы идентификации ресурсов URI, а сама аббревиатура URL постепенно уступает место более широкому обозначению URI.
URN — это идентификатор URI, который идентифицирует ресурс в определенном пространстве имен. Например, ISBN 0-395-36341-1 — это URI, указывающий на ресурс (книгу) 0-395-36341-1 в пространстве имен ISBN. В последнее время появилась тенденция говорить просто URI о любой строке-идентификаторе без дальнейших уточнений.
Web-сервер, получив запрос, разделяет его на части и выделяет идентификатор URI запрашиваемого ресурса. Далее по пути, указанному в идентификаторе URI-запроса, в файловой системе сервера находится запрашиваемый объект. В случае если для запрашиваемого объекта не указана программа выполнения (например, файл является HTML-документом или объектом мультимедиа), web-сервер возвращает его в теле HTTP-ответа. Если для запрашиваемого объекта указана программа выполнения (например, файл является скриптом на языке perl, php, ruby или python) либо если объект сам является исполняемым файлом, выполняют следующие действия:
• инициируют выполнение объекта, указанного в URL. Входными данными для этого объекта являются параметры HTTP-запроса. Правила доступа к параметрам HTTP-запроса из кода исполняемого объекта определяются конкретной технологией сопряжения Web-сервера с Web-приложением;
• исполняемый объект динамически генерирует тело HTTP-ответа в зависимости от входных параметров;
• сгенерированный выполненным объектом HTTP-ответ возвращается клиенту и обрабатывается браузером стандартным образом;
• сервер отвечает сообщением, содержащим строку статуса (включая версию протокола и код статуса — успех или ошибка), за которой следует сообщение в формате MIME, включающее в себя информацию о сервере, метаинформацию о содержании ответа и само тело ответа.
Большим шагом в развитии технологий генерации динамического контента был выпуск платформ Java Enterprise Edition (J2EE) и Microsoft .NET. Так же как и в языке Java, среда разработки .NET создает байт-код, предназначенный для исполнения виртуальной машиной. Применение байт-кода позволяет достигать независимости от конкретной среды выполнения (платформы). Перед запуском сборки программной системы в среде исполнения (Common Language Runtime — CLR) ее байт-код преобразуется встроенным в среду JIT-компилятором в машинные коды целевого процессора.

Интерфейс CGI (Common Gateway Interface) — это один из первых стандартов сопряжения web-серверов и программ. Основная задача такого сопряжения — определить, как передать программе параметры HTTP-запроса и как передать обратно сформированный программой HTTP-ответ. Интерфейс CGI реализует самый простой и очевидный способ: файл, указанный в HTTP-запросе (если он не является статическим ресурсом), запускается в виде отдельного процесса-обработчика. При этом:
• стандартный поток ввода процесса-обработчика ассоциируется с потоком вывода web-сервера;
• стандартный поток вывода процесса-обработчика ассоциируется с потоком ввода web-сервера;
• параметры HTTP-запроса передаются процессу-обработчику через аргументы командной строки и через переменные окружения.
Таким образом, разработчики программ для динамического формирования HTML-страниц при использовании интерфейса CGI никак не ограничиваются в выборе технологий разработки. Недостатком такого сопряжения является необходимость подготовки и запуска нового процесса на каждый HTTP-запрос, что является достаточно дорогой операцией во всех операционных системах.

Организация сеансов в протоколе HTTP
В соответствии со спецификацией протокола HTTP web-сервер не поддерживает постоянного соединения с клиентом, и каждый запрос обрабатывается как новый, т.е. без какой-либо связи с предыдущими, поэтому нельзя ни отследить запросы от одного и того же посетителя, ни сохранить для него переменные между просмотрами отдельных страниц. Для того чтобы различные запросы одного пользователя связать в единую логическую цепочку, используется механизм управления сессиями. При этом сама цепочка связанных запросов называется сессией. Информацию о сессии должен хранить браузер.
Для реализации механизма управления сессиями используются два дополнительных поля HTTP-заголовка: Set-Cookie и Cookie.
Схема взаимодействия клиента и сервера при использовании механизма Cookie следующая:

• клиент запрашивает какой-либо документ с сервера;
• сервер включает в заголовок ответа поле Set-Cookie, в котором сообщает клиенту информацию, предназначенную для сохранения, и параметры, задающие область действия этой информации;
• при каждом следующем запросе клиент возвращает серверу сохраненную информацию с помощью поля Cookie в заголовке запроса, если запрашиваемый ресурс попадает в область действия, заданную сервером;
• ресурс на сервере анализирует значение поля Cookie и идентифицирует клиента.
Первоначально механизм Cookie был описан в спецификации Netscape Communications Persistent Client State HTTP Cookie, в дальнейшем был принят документ RFC-2109 (HTTP State Management Mechanism).

Безопасность в HTTP: установление подлинности
Протокол HTTP предоставляет простой механизм аутентификации пользователя ресурсов web-сервера, построенный на обмене информацией между клиентом и сервером ресурса. Этот механизм позволяет выбрать схему аутентификации и уровень секретности передаваемых данных. Стандартом HTTP описываются две схемы аутентификации: Basic и Digest. Механизм аутентификации реализует web-сервер.
Схема Basic представляет собой простейший механизм аутентификации, в котором браузер пользователя делает запрос к защищенной странице, а web-сервер возвращает указание на необходимость аутентификации:

HTTP/1.1 401 Authorization Required

Браузер пользователя запрашивает имя пользователя и его пароль через специальное диалоговое окно. Затем он объединяет полученные параметры аутентификации и кодирует полученную строку с помощью алгоритма base64, после чего повторяет запрос к странице, передавая сформированную строку в специальном заголовке запроса. Недостатком этой формы аутентификации является передача по сети имени пользователя и его пароля в открытом виде.
Схема Digest предназначена для того, чтобы дать пользователю возможность доказать серверу, что он знает правильный пароль без передачи его по сети. Для этого в качестве параметра аутентификации серверу передается результат применения алгоритма MD5 к строке, состоящей из имени пользователя, его пароля, адреса запрашиваемой страницы и специального счетчика запросов, который хранится на сервере. При каждом новом запросе значение счетчика увеличивается на единицу. В результате данные об имени и пароле пользователя передаются по сети зашифрованными стойким шифром. Включение счетчика запросов в хэшируемую по MD5 строку позволяет защититься от атак повторением.
Программист web-приложения может отказаться от стандартных схем аутентификации и реализовать свою схему. Самый распространенный на сегодняшний день вид аутентификации — это аутентификация через формы. В этом случае логин и пароль пользователя вводятся в специальные поля web-приложения и передаются как обычные параметры в запросах POST или GET, а параметры аутентификации обрабатываются не web-сервером, а самим web-приложением. Преимущество такой схемы — это возможность реализации гибкой системы управления пользовательскими учетными записями и ролей пользователей.

Безопасность в HTTP: обеспечение конфиденциальности и целостности
Для обеспечения шифрования данных при взаимодействии двух систем по протоколу HTTP обычно используется протокол SSL, основанный на криптографии с открытым ключом. Протокол SSL обеспечивает три основных механизма зашиты: взаимную аутентификацию, обеспечение конфиденциальности и целостности данных. На этапе взаимной аутентификации сервер передает клиенту свой сертификат, а клиент может передавать серверу свой. Сертификат представляет собой документ в электронной форме, содержащий открытый ключ, принадлежащий его держателю, и дополнительную информацию о его владельце (например, ФИО владельца и название организации, где он работает, адрес электронной почты и т. п.), который подписан удостоверяющим центром (Certificate Authority). Основная задача сертификата — связать открытый ключ с личностью его владельца (владельца парного ему закрытого ключа). Сертификаты имеют срок действия, по окончании которого они становятся недействительными.
Срок действия отражен в содержании сертификата. Получив сертификат сервера, клиент проверяет его соответствие открытому ключу сервера, а сервер, в свою очередь, проверяет сертификат клиента, если он был передан. Использование сертификатов гарантирует, что клиент связан именно с требуемым ему сервером, а сервер точно знает, кто к нему подключился. В случае небольших групп пользователей механизм сертификатов может использоваться для аутентификации пользователей и авторизации их доступа к ресурсам сервера. Весь обмен сообщениями между SSL-сервером и SSL-клиентом шифруется с помощью ключа и алгоритма шифрования, которые согласовываются во время начального SSL-сеанса.
Перед установкой SSL-соединения протокол проверяет целостность данных (при шифровании она сохраняется автоматически). Для реализации службы целостности сообщений в протоколе SSL используется комбинация алгоритма шифрования с общим закрытым ключом и хэш-функцией.

Билет № 70.
Веб-технологии на стороне клиента: HTML, DOM, CSS, JavaScript (AJAX), Java-апплеты, Flash. Same Origin Policy.

Язык разметки HTML
Для представления документов на Web-серверах и связывания их между собой используется специальный язык разметки — Hypertext Markup Language (HTML). Этот язык не относится к алгоритмическим языкам программирования. Файл на языке HTML, являющийся документом,
содержит набор данных и правил их отображения; какие использовать шрифты, цвета, какие данные заключить в таблицу, где поместить изображение, видео, какого цвета будет фон и т.д. За загрузку документов с Web-серверов, интерпретацию языка разметки и отображение информации пользователю отвечают браузеры. Язык HTML решает задачи унифицированного представления документов в разных операционных системах на разных платформах и в разных сетях, обеспечивая их широкую доступность. Вначале HTML был просто языком форматирования гипертекстовых документов. Элементы формата ограничивались заголовками, абзацами и небольшим набором форматов текста, таких как полужирный шрифт или курсив. Такие элементы формата называются тегами, представляющими собой ключевые слова, заключенные в угловые скобки. Стандарт HTML 4.0 добавил к первоначальному языку ряд новых мощных функций и превратил ограниченный язык задания формата в полноценный инструмент структурирования. Язык HTML 4.0 был стандартизован консорциумом W3C, что существенно ограничило деятельность компаний (Microsoft, Netscape) по расширению его спецификации своим синтаксисом.

Каскадные таблицы стилей CSS. Основной задачей стандарта HTML 4.0 было отделение структуры и текста документа от его представления и стиля. Таким образом появились стили и язык описания стилей CSS (Cascade Style Sheets). Идея использования таблиц стилей CSS очень проста. Если раньше в HTML необходимо было прямо в документе указывать, как должен выглядеть тот или иной элемент, то при использовании описаний CSS такие указания выносятся в отдельный блок, который может загружаться в виде отдельного файла (типа include).
Очевидны следующие преимущества такого подхода. Во-первых, значительно облегчается изменение внешнего вида сайта и отдельных его элементов, т.е. достаточно изменить определение соответствующего стиля в единственном CSS-файле и эти изменения распространятся на весь сайт. Во-вторых уменьшается размер документов, что особенно заметно на ≪красивых≫ страницах, а это способствует их скорейшей загрузке на клиентские машины.

Набор технологий DHTML для создания интерактивных страниц.
По мере развития WWW стало очевидно, что концепция о неизменности HTML-страницы после ее загрузки с сервера существенно ограничивает возможности представления и логику обработки информации. Например, эта концепция не поддерживала ≪выпадающие≫ меню. Кроме того, вся логика по обработке информации реализовывалась на сервере, значит, для проверки корректности данных, введенных клиентом в формы, необходимо было отправить запрос на сервер, что существенно замедляло скорость работы.
Перечисленные недостатки HTML стали предпосылками появления концепции DHTML (Dynamic HTML), которая не была оформлена в виде стандарта и которую не следует рассматривать как новую спецификацию языка HTML. В концепции DHTML определяется набор технологий, позволяющих браузеру динамически изменять загруженные HTML-документы в ответ на пользовательские действия без взаимодействия с Web-сервером. Таким образом, часть логики работы с HTML-документом выносилась на сторону клиента, т. е. на абонентскую машину.
Основу концепции DHTML составляют следующие основные компоненты: язык HTML, язык описания каскадных таблиц стилей CSS, скриптовый язык, который интерпретируется браузером (например, JavaScript), и объектная модель HTML-документов — DOM.
Язык HTML был расширен конструкциями специального типа — событиями. Так, с любой структурной единицей документа можно связать событие и обработчика этого события. Например, для некоторой таблицы можно указать, что по нажатию пользователем левой кнопки мыши над этой таблицей, она окрасится в цвет, код которого введен в специальное текстовое поле на этой же странице. Обработчики таких событий реализуются с помощью специального кода, называемого скриптом, который интерпретирует браузер. Самым распространенным языком реализации этих обработчиков является JavaScript. Однако остается вопрос: как программа на языке JavaScript сможет поменять структуру или представления загруженного HTML-документа? Ведь в этом случае браузер должен предоставить обработчику некоторый интерфейс для доступа к HTML-документу. Браузеров много, и вряд ли их производители договорятся о едином интерфейсе.

Поэтому консорциум W3C разработал стандарт, т.е. объектную модель HTML-документа — DOM (Document Object Model) и интерфейс, который должен использоваться для работы с этой моделью.

Технология асинхронного взаимодействия с web-сервером — AJAX

Пусть у пользователя есть на некотором сайте фотоальбом в режиме доступа on-line. В соответствии с логикой функционирования при последовательной навигации по своему альбому пользователю необходимо перегружать только фотографии, а не всю страницу.

Концепция асинхронного взаимодействия с web-сервером — AJAX (Asynchronous JavaScript and XML). Основой технологии AJAX является DHTML, а главная ее задача состоит в уменьшении числа обращений к серверу. При обновлении информации AJAX-приложение не перегружает страницу полностью. Вместо этого код на языке JavaScript отправляет XML-запрос на сервер, а затем заменяет необходимую часть страницы, используя интерфейс модели DOM для обновления. Широко известным примером AJAX-приложения является Google™ Maps. Асинхронный обмен данными более удобен для работы, поскольку пользователю не приходится смотреть в пустой экран, дожидаясь перезагрузки страницы. В идеале пользователь вообще не должен замечать, когда приложение обратилось к серверу, так как данные подгружаются в фоновом режиме мелкими порциями.

Java-апплеты и технология Flash

Концепция вынесения логики работы с Web-сервера на сторону клиента не остановилась в своем развитии на концепции DHTML. Дело в том, что JavaScript и другие языки, интерпретируемые браузером, имеют ограниченные возможности по сравнению с интерпретируемыми языками общего назначения, например такими как Java. Понимая это, компания Sun Microsystems разработала технологию Java-апплетов. Java-апплет — это прикладная программа на языке Java в форме байт-кода. В отличие от программ на языке JavaScript, которые интерпретируются браузером, Java-апплет выполняет виртуальная Java-машина (JVM), которая и является интерпретатором байт-кода и которую устанавливают на хосте отдельно. Java-апплеты были внедрены в первой версии языка Java в 1995 г.

Java является языком общего назначения, а следовательно, необходимо предпринимать специальные меры по обеспечению безопасного выполнения Java-апплетов на компьютерах пользователей, ведь загруженный из Интернета код вместе с HTML-страницей может обладать как полезными, так и вредными функциями. Для обеспечения безопасности пользователей апплеты выполняются в специальной песочнице (sandbox), которая ограничивает взаимодействие байт-кода Java с окружением: запрещает операции считывания и записи файлов, запуск других приложений на компьютере пользователя, а сетевой доступ из апплета допускает только к тому хосту, с которого он был загружен.

Апплету разрешено считывать значения параметров (цвета, шрифты, файлы с графическими изображениями и т.д.) с содержащей его Web-страницы и в соответствии с этими параметрами изменять свое поведение. Кроме того, параметры апплета можно изменять динамически.

Альтернативной технологией для вынесения логики на сторону клиента стали приложения Adobe Flash. Flash-приложения реализуются на языке ActionScript, являющемся объектно-ориентированным языком программирования и одним из диалектов стандарта ECMAScript, который добавляет интерактивность, обработку данных и многое другое в содержимое этих приложений. Язык ActionScript компилируется в байт-код и исполняется виртуальной машиной ActionScript Virtual Machine. В основе технологии Flash лежит векторный морфинг, т.е. плавное ≪перетекание≫ одного ключевого кадра в другой, что позволяет делать достаточно сложные мультипликационные сцены, задавая лишь несколько ключевых кадров для каждого персонажа.

Технология Flash в основном используется для написания игр, небольших полуинтерактивных анимаций и для красивого оформления рекламы, т.е. в сфере развлечений и дизайна. Для серьезных Web-приложений, где взаимодействие с пользователем должно быть без ущерба красоте, обычно используется язык JavaScript либо вообще не используются никакие технологии кроме тех, которые работают при любых настройках безопасности в браузерах (HTML, CSS).