**Лекция за 28.03.2012, часть 2**

***Протокол Frame-Relay.***

***Что же это такое?*** FR также предназначен для передачи данных в глобальных сетях, но в отличие от ATM и синхронных иерархий, первоначально он вообще был рассчитан на соединение точка-точка, а в дальнейшем появились FR-коммутаторы, которые поддерживали топологию звезды. ***Основная задача FR – решение «проблемы последней мили»***. Когда вам необходимо передать какие-либо данные в том случае, когда непосредственно строить канал передачи данных либо дорого, либо физически невозможно. Первоначально FR, собственно, откуда его название и проистекает, предназначался для передачи данных по радиоканалу. В дальнейшем он получил возможность работать по любому каналу связи, в т.ч. и по проводному.

Что из себя представляет FR? Лучше называть это пакетом, чтобы нас лучше понимали.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| StartFlag | Format/Address | Data | CRC16 | StopFlag |
| 01111110 | 2-4 | to 1600 | 2 | 01111110 |

***Технология FR использует пакеты переменной длины***. Причем длина пакета нигде не фиксируется, границы пакета определяются только на основании стартовых и стоповых флагов. Фактически так же, как это осуществляется в современных Ethernet-технологиях.

***В качестве стартового и стопового флагов используется последовательность из шести единиц, ограниченных двумя нулями***. Чтобы не перепутать ничего со стартовым и стоповым флагом, если где-то при передаче данных внутри пакета FR встречается последовательность из пяти единиц вне зависимости от того, что будет дальше, в нее добавляется дополнительный ноль. При приеме данных последовательность “пять единиц ноль” транслируется просто в пять единиц. Как легко понять, процедура однозначна и избавляет от появления шести единиц подряд. В ethernet-е механизм несколько более сложный, он построен на ?треллис?-кодировании, при котором последовательность из 6 или 7 единиц просто не появляется никогда.

Дальше следует поле, которое называется ”***Format/Address***”, ***оно занимает от 2 до 4 октетов***. Собственно говоря, длина поля определяется первыми битами этого поля. Поскольку первое поколение сетей FR это вообще только соединение точка-точка, то там вообще никакая адресация не требовалась. В сетях второго поколения FR, когда появилась организация посредством звезды, там все равно было какое-то конечное количество абонентов, т.е. коммутаторы FR были на три, на четыре, ну может быть на восемь подключений. Решили не вводить специальное поле для адреса и оставили просто последние биты этого поля “***Format/Adress***” в качестве адресных полей(они там все равно не использовались). ***Соответственно на адрес идет 4 бита из этого поля. Последние 4 бита вне зависимости от того, какой длины это поле вообще. Т.о. FR позволяет использовать до 15 адресов. Адреса из одних нулей соответствуют первому поколению FR, соединению точка-точка***.

Следующее поле это ***поле данных***. В первом поколении ***поле данных произвольной длины до 1600 октетов*** (подгонялось, чтобы туда без проблем инкапсулировался Ethernet-овский трафик с максимальной длиной фрейма). ***Во втором поколении расширили до 4096 октетов***. В основном опять же для того, чтобы уменьшить overhead (перегрузку) на передачу заголовка, а также из соображений устойчивой передачи. Оказалось, что оборудование FR особенно по проводным линиям работает с достаточно большим запасом и позволяет использовать большую длину поля данных. Реально максимальная длина поля данных настраивается на оборудование FR. Важно не смешивать оборудование первого и второго поколения.

Дальше идет ***поле контрольной суммы***, ***вычисляемой по алгоритму*** ***crc16,*** и дальше идет стоповый флаг. Соответственно ***стартовый и стоповый флаги не смешиваются***. То есть есть технологии, когда стоповый флаг предыдущего фрейма может служить стартовым для последующего, так вот здесь это не так: каждый фрейм должен быть строго ограничен с двух сторон. Это вообще довольно распространенная идея, которая используется во многих сетях при передаче по радио (а первоначально FR создавался именно для беспроводной связи).

Первая линия FR появилась в 1996 году, соединяла Университет с городом. Тарелка стояла на НИИФе, а вторая стояла на ТЭЦ рядом с платформой Броневая.

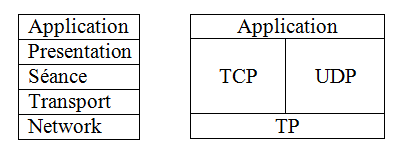
***Протоколы TCP/IP.***

Первые исследования были сделаны в конце 1960х годов, и никто не предполагал, что из протоколов этого семейства мы получим основной транспортный сетевой протокол сети Интернет. С этим связаны многие проблемы TCP/IP. Изначально он возник как академическая разработка Калифорнийского университета в Беркли. Неудивительно, что там появились и первые практические реализации этого протокола. В качестве референсного стека TCP/IP считается, что впервые стал использоваться сетевой стек, который был включен в ОС BSD 4.2. Этот релиз также получил название BSD.Net, поскольку это был первый релиз коммерческой ОС с поддержкой сетевого стека общего назначения.

На самом деле надо просто понять, что в 1960е-70е годы компьютерными разработками в США занимались люди, которые вообще-то исповедовали идеологию, находящуюся на стыке между такими идеологиями как либеральная, либертарианская, и вообще полным анархизмом. Поэтому на самом деле тогда очень активно развивалась точка зрения, которую можно назвать “пусть растут все цветы” по соответствующему лозунгу хиппи. То есть тогда считалось, что глобальная сеть будет чрезвычайно разнородной, что она будет включать в себя большое количество физических протоколов, сетевых, транспортных, и предполагалось, что достаточно долгое время они будут конкурировать друг с другом, будут существовать различные реализации и никто вообще не думал, что не пройдет и тридцати лет и мы практически во всех технологиях физического уровня будем использовать Ethernet (то есть даже в технологии глобальных сетей сейчас наблюдается переход к Ethernet-технологиям), а в технологиях сетевого уровня это будет протокол IP.

Следующим важным этапом после релиза BSD.net можно видимо считать появление ОС Windows 3.11 for workgroups (или как тогда ее называли графическая оболочка DOS), которая впервые для ОС Windows получила поддержку протокола IP. На самом деле это сейчас Microsoft такой белый и пушистый и заявляет о поддержке открытых стандартов. А на самом деле в тот момент Microsoft был очень молодой и агрессивной компанией, они всерьез считали, что все ходят не в ногу, они одни только ходят в ногу и от них тогда слышались заявления, что Microsoft никогда поддерживать IP не будет, что у них есть гораздо лучшие решения. Но, однако, когда конкуренты подсуетились и начали предлагать на базе систем MS-DOS и Windows решения на основе TCP/IP, Microsoft довольно быстро взялся за ум и т.к. разрабатывать свой стек с нуля им было некогда, они купили компанию ?Banyan Vines? и использовали в ОС Windows for workgroups и Windows 95 стек ? Banyan Vines?. Дальше они развивали тот стек, который был положен в основу ОС Windows NT и который в той или иной форме дошел до настоящего времени в современных ОС, как десктопных, так и серверных компании Microsoft. Какая-то доля стека ? Banyan Vines? там видимо присутствует, а сам стек ? Banyan Vines? был сделан на основе стека BSD.net. Почему за основу брался BSD.net тоже понятно – дистрибуция исходных кодов ОС BSD велась под лицензией BSD, которая позволяла довольно свободно модифицировать исходные коды, исходные тексты программ, а главное, она позволяла делать на их основе закрытые разработки, с закрытым исходным кодом. Хотя она формально требует, чтобы закрытый готовый продукт также дистрибутировался по крайней мере с включением лицензии BSD, но она не требует лицензии на весь продукт, что позволяет на основе BSD разрабатывать закрытый кросс-лицензионный код.

Что из себя представляет стек TCP/IP? Мы будем считать, что у нас на физическом уровне либо Ethernet, либо тот же ATM. Соответственно, стек TCP/IP первоначально выглядел очень просто.



***Сетевому уровню ISO/ASI соответствовал уровень IP***. Первоначально, IP расшифровывался как Interchange protocol (протокол обмена), но в дальнейшем, с появлением терминологии Интернет, расшифровка была изменена на internet protocol.

***Следующие*** ***три уровня ISO/ASI: транспортный, сеансовый и представления, занимали два протокола, которые в терминологии стека TCP/IP носят название транспортного уровня, это протоколы TCP и UDP***. ***TCP – transport control protocol – протокол контроля передачи и User Datagram Protocol – протокол пользовательских датаграмм***. ***Уровень приложений соответствует уровню приложений ISO/ASI***. То есть, по первоначальному представлению, уровень TCP/UDP соответствует трем уровням представления модели ISO/ASI. Это одна из причин, по которой в свое время модель ISO/ASI критиковалась. Есть протокол TCP/IP, там хватает одного уровня TCP, а в модели ISO/ASI целых три.

Но в дальнейшем модель ISO/ASI стала все больше и больше привлекать внимание масс, приобретала все большую популярность, и выяснилось, что такой уж заметной разницы между стеками TCP/IP и ISO/ASI вроде как и нет. Сейчас обычно эту модель представляют следующим образом

|  |  |
| --- | --- |
| Application | |
| Sockets  TCP | Sockets UDP |
| /////////////////// |
| TCP | UDP |
| IP | |

Уровень представления теперь занял уровень сокетов. ***Сокет это такая особенная структура данных, которая включает IP и работу с ними и чем-то напоминает пайпы, которые существуют в локальных файловых системах и предназначены для взаимодействия процессов передачи данных между ними. То есть любой процесс может записывать данные в сокет, а удаленная сторона может данные с сокета читать. Сокеты выполняют основную функцию, которая возложена на уровень представления, а именно уровень сериализации-десериализации пользовательских данных***. На самом деле в связи с тем, что существуют отдельно сокеты TCP и сокеты udp, то соответственно граница между сокетами TCP проходит примерно посередине сеансового уровня, то есть часть функции сеансового уровня выполняет непосредственно протокол TCP, а часть – непосредственно IP-сокеты. Единственная оставшаяся разница заключается в том, что при использовании транспортного протокола UDP функции сеансового уровня оказываются нереализованными. Протоколы UDP вообще не используют такого понятия, такой терминологии как сеанс.

***В чем разница между протоколами TCP и udp. Протокол TCP обеспечивает установление соединения и контроль целостности передаваемых данных. Протокол udp соединение не устанавливает и целостности данных не обеспечивает***. Если вам требуется поддержка целостности данных с использованием протокола udp, то эту задачу должен решать протокол пользовательского уровня. Для чего применяются тот и другой тип транспортного протокола? ***TCP используется там, где требуется надежная доставка, а udp - там, где требуется доставка в реальном времени, поскольку задержки при передаче данных у протокола TCP довольно высоки***. Основная задача протокола udp – доставка мультимедийных данных, потокового аудио и видео. Также поверх него реализуются такие решения, как IP-телефония и т.п. В принципе поверх него можно реализовывать решения с целостностью данных, например с помощью протокола udp работала сетевая файловая система unix. Там целостность данных поддерживалась на уровне пользовательского протокола. Хотя в настоящее время система NFS5 использует в качестве транспортного протокола протокол TCP. Связано это с тем, что в настоящее время многие проблемы обработки протокола TCP ушли в прошлое и по современным высокоскоростным сетям возможна передача TCP без существенных задержек.

На самом деле, если вдаваться в историю. Существовал еще один транспортный протокол tccp (transport connection control protocol), но от его реализации ввиду его сложности в релизе bsd.net решено было отказаться в надежде на то, что если он кому-то понадобится, его реализуют. Но оказалось, что все, что можно сделать с его помощью, можно сделать и с помощью протокола TCP. На самом деле единственное отличие от TCP только в том, что он в явном виде реализовывал понятие сеанса. С точки зрения первоначальной логики, отличий от первоначальной логики ISO/ASI было еще меньше.

Каким образом осуществляется адресация в протоколах семейства IP***. IP-адрес – это совокупность 4 чисел в десятичной записи, каждое из которых в диапазоне от 0 до 255***. То есть в принципе каждую компоненту адреса можно записать в виде двух шестнадцатеричных цифр. То есть фактически размер каждого числа адреса это один октет. Поэтому довольно часто компоненты IP адреса называются октетами. Я не рекомендую использовать это термин, т.к. мы постоянно октетами называем размер передаваемых данных длиной в 8 бит. ***Соответствующая нотация называется десятичной или тетрадной (правильнее ее было бы называть восьмерично-десятичной), поскольку адрес состоит из 4 групп***. Другая альтернативная форма записи она естественно в практических целях не используется, но используется в теоретических работах, это двоичная запись в виде единого 32битного двоичного числа.

***Адрес состоит из 2 логических частей – сетевого адреса и адреса хоста в сети. Сетевой адрес адресует сеть, а адрес хоста адресует хост внутри отдельной сети***. В связи с этим выделяется 2 специальных типа адреса. Первый – широковещательный адрес внутри сети. Этот адрес состоит из одних единиц на месте адреса хоста. Второй специальный адрес, состоящий из одних нулей на месте адреса хоста, изначально предполагалось использовать при доступе снаружи. Т.к. широковещательные запросы в ethernet-е представляли очень большую проблему, то эти адреса перестали маршрутизироваться, и перестали использоваться. Кроме того существует еще один специально выделенный адрес (спец выделенная сеть), обычно это адрес 127.001. Это так называемый адрес lookback-интерфейса. Что это такое? Любой хост, на котором поднят стек TCP/IP имеет специальный виртуальный интерфейс, даже если он не имеет ни одного физического сетевого интерфейса и этот виртуальный интерфейс имеет адрес из этой сети. То есть первоначально этот интерфейс использовался в целях тестирования, а в дальнейшем в целях совместимости был сохранен и по-прежнему используется для этих целей.

***Теперь о планах адресации и нумерации в сетях IP***. Понятно, что если есть глобальная сеть, то должно быть глобальное пространство имен. Первоначально, когда об интернете не было еще никаких мыслей, а существовала сеть МО США. Это первая сеть, в которой использовались протоколы TCP/IP. Именно тогда было решено отделить сетевую и хостовую часть адресов по границам октетов, то есть по границам 8битных интервалов исходного 32битного адреса. Возникла классовая схема адресации.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | группы в тетради (сеть/хост) | сетевые адреса | хостовые адреса |
| A | 0… | 1/3 | 126(128) | 16777216 |
| B | 10… | 2/2 | 16384 | 65536 |
| C | 110… | 3/1 | 2097152 | 256 |
| D | 1110… | - |  | 268435456 |
| E | 1111… | - |  | 134217728 |
| F | 11111… | - | - | 134217728 |

По разным классификациям вводилось либо 5, либо 6 классов сетей, они различались по первым битам двоичной записи адреса и при этом различались по тому, сколько октетов использовалось для записи адреса в сети и адреса хоста.

***Сети класса A имели лидирующий бит нулевой и использовали одну группу в тетради для записи адреса сети и три группы для записи адреса хоста***. Соответственно, максимальное количество ***сетевых адресов могло быть 128, на самом деле 2 из них выпадали***: адреса, начинающиеся с нулевого бита, в адресе сети были зарезервированы, а во вторых, адреса, начинающиеся со 127. Соответственно, ***хостов могло быть 16777216, правда, опять же минус 2***, потому, что адреса из одних нулей и одних единиц в поле адреса хостов невозможны. Это адреса широковещательной рассылки.

***Сети класса B начинались с лидирующих '10' и имели 2 группы в тетради для адреса сети и 2 для адреса хоста. Было n адресов сети и m адресов хостов. Ну, опять же, -2 по той же самой причине***.

***Сети класса C имели лидирующие адреса 110 и имели три группы в тетради для адреса сети и одну для адреса хоста. Что давало примерно 2097152 адресов сети и 254 адреса хостов в каждой сети. Эти 3 группы ABC предназначались для публичной адресации***.

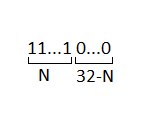
***Предполагалось, что крупные компании будут иметь адреса в A, средние B, мелкие – C***. Под крупными компаниями понимались прежде всего крупные государственные учреждения, такие, как пентагон, Минздрав, почта и такие крупные компании как GM, IBM. К классу B относились более мелкие компании, не им представительств на всей территории США, ну а все остальные относились к классу С. ***Сети класса D – их части не делились на сетевую и хостовую часть и начинались с 1110. Всего их могло быть - 268435456. Эти адреса предполагалось использовать для так называемой мультикастовой рассылки***. Адреса можно было использовать на нескольких хостах. Если этот адрес стоял в поле адреса получателя, то этот пакет доставлялся всем хостам имеющим данный адрес.

Из этого ничего не вышло, т.к. разработанные на тот момент алгоритмы маршрутизации не имели нормально разработанной поддержки мультикастовой маршрутизации и что более важно, непонятно было как указывать адрес отправителя, если он мультикастовой***. Протокол TCP принципиально двунаправленный – если данные кому-то пришли, то надо отвечать***. Получалось, что эти хосты должны были иметь еще и обычные адреса из того же класса ABC. Терялось преимущество уменьшения расходов IP-адресов.

***Классы E,F иногда объединяют в один класс, они ни для какой цели не использовались, были зарезервированы***. Классовая адресация была очень логичной, достаточно просто алгоритмически реализуется, поскольку выделять куски в 32битном слове по 8битной границе достаточно просто.

***К сожалению, эта схема очень быстро напоролась на проблему исчерпания IP-адресов, потому, что в классе A очень много адресов остались незанятыми и в классе B тоже, а класса C стало очень быстро не хватать***. Поэтому на самом деле уже в 1994 году была проведена уникальная в своем роде операция, а именно изменение адресной политики во всей северной Америке, тогда было решено отказаться от классовой адресации, а поскольку классики марксизма-ленинизма учат нас, что мы движемся в направлении бесклассового общества, то и соответствующую адресацию было решено назвать бесклассовой.

***Основная задача бесклассовой адресации заключается в том, чтобы отказаться от деления на сетевую и хостовую часть адреса по границе октетов, и дать возможность разделения адресов по любому биту адреса. Это решает еще и проблему увеличения объема маршрутных таблиц. Для того чтобы дать возможность разделить сетевую и хостовую часть по любому биту, было введено еще одно понятие, а именно маска подсети. Маска подсети – число в двоичной записи, которое содержит 1 во всех полях адреса, которые относятся к адресу сети***. Число из n единиц записывается в двоичной системе как 2^n-1. Т.к 2^n это единица и n нулей, когда мы вычитаем единицу, остается n единиц.

  
 ***Если у нас в сетевой части адреса n единиц, то двоичная запись маски подсети будет . С этим связана упрощенная форма записи адресов в случае бесклассовой адресации, т.е. если мы используем маску подсети из n единиц, то запись a.b.c.d/N означает, что этот адрес имеет сетевую часть длиной N бит***.

Существует такая запись маски подсети также в виде тетради или 4 десятичных чисел, по полной аналогии с IP-адресом. Соответственно маски подсети для классов ABC должны выглядеть следующим образом:

***255.0.0.0 для класса A***

***255.255.0.0 для B***

***255.255.255.0 для C***.

Если вы будете сисадминами или IP-программистами, то вам придется выучить маски подсетей, по крайней мере, внутри класса C.

|  |  |
| --- | --- |
| размер маски |  |
| /24 | 255.255.255.0 |
| /25 | 255.255.255.128 |
| /26 | 255.255.255.192 |
| /27 | 255.255.255.224 |
| /28 | 255.255.255.240 |
| /29 | 255.255.255.248 |
| /30 | 255.255.255.252 |

Это наиболее распространенные сети, которые выдаются небольшим компаниям. Соответственно сам класс C(/24) имеет на конце ноль. (128+64=192, 192+32=224, 224+16=240, 240+8=248, 248+4=252). Маски /31 и /32 в принципе невозможны. Почему невозможна маска 32 понятно, потому, что 32 бита не может соответствовать ни одному сетевому адресу. Иначе адрес должен состоять из 32 бит, и на хост ничего не останется. Маска 31 тоже существовать не может, потому что в этом случае в сети будет всего 2 адреса, и на адрес хоста будет уходить всего 1 бит, но адрес хоста из одних нулей или одних единиц тоже не бывает, потому, что это широковещательные адреса. В результате чего максимальный размер маски это 30. Введение бесклассовой адресации позволило более гибко выделять адреса для особенно крупных компаний и бороться с проблемой огромных неиспользуемых адресных пулов.

Второе направление борьбы – введение специальных немаршрутизируемых адресов. Это адреса, которые предоставляются частным сетям, которые не имеют выхода в Интернет. Первоначально в каждом классе ABC было выбрано по одной подсети, для использования в этих целях. Понятно, что если у вас использовалась сеть класса ABC и вам не требуется выход в Интернет, то вам проще всего поменять просто первый октет для класса A, первый и второй для класса B, и первый, второй, третий для класса C. В соответствующих классах A,B,C это сети: 10.0.0.0, 172.16.0.0, 192.168.(0..255).0. Все эти сети не должны маршрутизироваться в Интернете. Потом были добавлены еще дополнительные сети, например сеть 169.16.0.0 – используется для автоматически настраиваемых устройств.

Соответственно это позволило на долгое время решить проблему исчерпания IP-адресов, и в 1994 году, когда была операция по изменению маршрутизации в северной Америке, это был единственный за всю историю сети Интернет момент, когда Интернет был отключен и не работал. В это время администраторы узлов в течение примерно двух суток меняли настройки своих компьютеров, и Интернет по всей земле практически стоял. Более того, в США и Канаде это дало статистически значимый всплеск рождаемости. На самом деле следующее исчерпание IP-адресов прочили на прошлый февраль (февраль 2011). Но обошлось без глобальных действий.

В настоящее время добиться от провайдера, например выделения IP-адресов в классе C довольно сложно. По разным оценкам в настоящее время нераспределенными является где-то от 3 до 10 процентов адресного пула. Почему такой разброс в общем то понятно – т.к. часть сетей находится в подвешенном состоянии и про низ мало что известно. Это либо пулы выделенные провайдером и неизвестно, насколько широко они могут использоваться. То есть можно выделить полную сеть класса C, а оборудование провайдера поддерживает только 128 одновременных соединений. С другой стороны просто часть организаций, которым выделялись соответствующие адресные пулы, прекратила свое существование и эти адресные пулы надо как-то возвращать в общий доступ.

Но в целом задача стоит, и проблема тоже стоит, то есть IP-адресов просто физически не хватает. На самом деле разумный способ с этим бороться – переходить на протокол IPv6, но он тоже под собой имеет определенные проблемы. По крайней мере, за развитие IPv6 отвечает так называемый IP6 консорциум – это совокупность и производителей оборудования, и программного обеспечения, и вообще известных частных лиц, которые большой вклад в развитие Интернета внесли и являются авторами многих стандартов Интернета. Одна из вещей, о которых они договорились, что перед тем, как внедрять IPv6 в массы они должны в частности проработать юридические вопросы распределения адресов в сети. А это очень серьезный камень преткновения, т.к. распределение любых ресурсов это вещь потенциально денежная. Поэтому интересы очень многих участников этого консорциума кардинально в этом вопросе различаются.

Что будет дальше, непонятно. Многие оценки сходятся только в одном – что где-то во второй половине 2010х годов IPv6 так или иначе придется внедрять. Особенно это связано с массовым появлением на рынке смартфонов, которые уже требуют подключения к Интернету, причем подключения к Интернету постоянного, как правило. И, собственно говоря, безусловно, есть технологии, которые позволяют получать данные из Интернета и для адресов, входящих в закрытые пулы (это так называемые неанонсированные адреса), но если вы хотите предоставлять свои данные в Интернет, а развитие технологий мобильной связи к этому идет, требуется как-то решать вопрос с адресным пространством. Наиболее разумное решение это внедрение протокола IPv6.