**Лекция за 28.03.2012, часть 1**

***Технологии глобальных сетей.***

Сегодня мы заканчиваем нашу тему, связанную с передачей данных на физическом и канальном уровне. В качестве финала у нас будут ***технологии глобальных сетей***. Во-первых, разделение сетей на глобальные и локальные условно, потому что те технологии, о которых пойдёт речь сегодня – или по крайней мере часть из них – либо применялась когда-то, либо легко адаптируема и для локальных сетей тоже. Разделение на локальные и глобальные во многом административное. Во-вторых, гораздо важнее, чем разница между локальностью и глобальностью – их функциональное предназначение, которое и определяет во многом принципы и подходы, используемые при проектировании соответствующих сетей.

Первая и, пожалуй, основная особенность глобальных сетей – ***разнородность передаваемых данных***. Кроме компьютерных данных, которые превалируют в локальных сетях, глобальные очень часто используются для передачи цифровой телефонии. Выросли они из телефонных цифровых сетей, которые начали внедряться во второй половине 60-х годов. Многие подходы определены именно этим. Большое количество мультимедийных данных связанных с передачей мультимедийного контента. Это уже появилось в последние годы и связано с использованием цифрового телевидения и превращением современных домашних компьютеров в производительные мультимедиа-центры. Вот два фактора, влияющих на глобальные сети – административный вопрос и функциональная нагрузка (контент).

С ***административной точки зрения*** глобальные сети отличаются от локальных. Глобальные сети дороги – это междугородные волоконно-оптические линии, подводные кабели и прочая инфраструктура, требующая больших вложений, которое могут позволить себе только компании-провайдеры глобальной связи. В России одна из таких компаний – Ростелеком. Среды передачи, как правило, сдаются провайдерами в аренду. Аренда – это либо гарантированная полоса пропускания (например, 10 Мбит/с), либо трафик определённого объёма в течение определённого срока (при этом скорость может гарантироваться или нет). Очень похоже на тарифообразование домашнего интернета 3-4 года назад. Сейчас основные тарифы – безлимитный Интернет с фиксированной скоростью, и в глобальных сетях ощущается то же направление. Но в планах тарифов и подходах глобальные сети более консервативны, потому что крупные компании всегда более консервативны, чем мелкие.

Глобальные сети возникли во второй половине 60-х и начали активно развиваться в 70-е годы для передачи цифровой телефонной связи в США, Японии и западной Европе. Это привело к созданию ***трёх независимых стандартов***. Забегая вперёд – в России глобальные сети стали активно строиться в 90-е годы в основном по американским стандартам. Это не значит, что в советское время не было собственных разработок, но они не вышли за пределы опытной эксплуатации и были практически повсеместно заменены технологиями на основе импортного оборудования.

Рассмотрим подход (самый простой, с которого всё началось), связанный с ***мультиплексированием данных***. Компания-владелец каналов организует ***виртуальные каналы*** и сдаёт их в аренду. Они имеют гарантированную ширину пропускания. Неважно, передаются в данный момент времени данные или нет. Могут использоваться арендатором по его собственному усмотрению, в частном случае никак. Такие технологии называются ***технологиями с мультиплексированием каналов*** (линий).

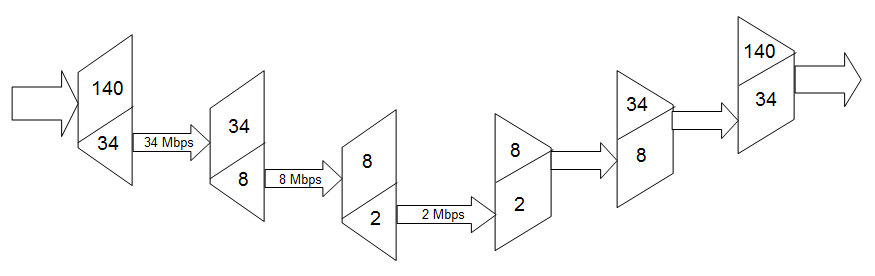


Рисунок . Схема мультиплексирования-демультиплексирования каналов

Представим, что у нас есть канал шириной 140 Мбит. Из него выделяется канал на 34 Мбит. 140 = 34\*4 + 4. Эти 4 либо передают служебную информацию, либо тоже сдаются в аренду. 34 = 8\*4 + 2. Делением каналов на мелкие занимается ***демультиплексор***. С обратной стороны линии – ***мультиплексоры***, объединяющие каналы в один. Картинка возникла из каналов телеграфной связи. Тогда провода физически объединялись в жгуты.

Кадры физического уровня в условиях использования виртуальных каналов рисуются в виде двумерной диаграммы.

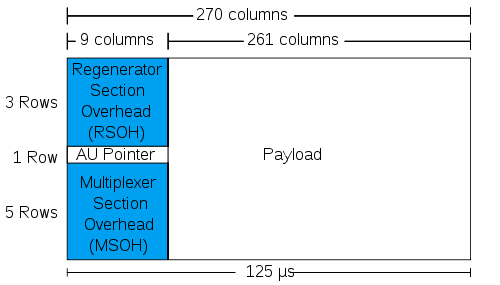


Рисунок . Кадр физического уровня STM-1

Эта картинка соответствует ***Synchronous Transfer Module-1*** (STM-1) - кадр протокола физического уровня в сетях с мультиплексированием данных. Картинка двумерная: в длину 270 октетов, в высоту 9 виртуальных каналов. Заголовок содержит три секции. *Регенеративная секция RSOH* (27 октетов) – для восстановления данных в случае возможных ошибок, действует только на заголовок. *Административная секция* (9 октетов) – соответствует тому, что в Ethernet содержится в заголовке канального уровня LLC. *Последняя секция MSOH* (45 октетов) - служебная нагрузка, которая определяет схему мультиплексирования-демультиплексирования каналов, т.е. какими реально абонентами используются виртуальные каналы, которые находятся в области полезной нагрузки. В некотором смысле эту секцию можно сравнить с секцией информации физического уровня – MAC-адресами.

Каким образом передаётся кадр? Вполне линейно, по строкам – сначала данные первого виртуального канала, потом второго, третьего и т.д. Соответствующие данные заголовка оказываются равномерно размешаны в общем потоке данных. Как мультиплексируются-демультиплексируются данные? У нас 9 виртуальных каналов, они могут принадлежать от 1 абонента до 9 абонентов. Понятно, когда каналы принадлежат разным абонентам – схема простая. Каждый абонент получает свой ***таймслот***, в течение которого передаются его данные. Что, если два виртуальных канала принадлежат одному и тому же абоненту? Передача всё равно происходит независимо. Сначала передаются данные первого виртуального канала, потом все остальные по очереди, потом следующий набор данных первого канала.

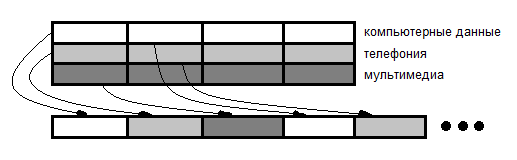


Рисунок . Порядок передачи данных в таймслотах (для упрощения три канала)

Данные каналов идут ***строго последовательно***, даже если принадлежат одному владельцу. Соответственно, владелец может по каждому из виртуальных каналов передавать разнородные данные (по одному компьютерные, по другому телефония, по третьему мультимедиа). Даже если никаких данных на каком-то канале в данный момент нет (например, телефонный разговор не ведётся), всё равно таймслоты будут выделяться. В этом ***самый большой недостаток*** синхронных мультиплексирующих схем. Из-за того, что они жёстко выделяют таймслоты для каждого абонента, получается огромный оверхед за счёт того, что предоставляется канал, по которому не передаются данные. Все дальнейшие работы велись по решению этой важной проблемы - как обеспечить возможность использовать неиспользуемую полосу пропускания.

Самое простое решение было найдено почти сразу – ***внутреннее мультиплексирование***. Если есть два канала данных, принадлежащих одному абоненту, он может перебрасывать данные между ними, используя ту часть пропускной способности, которая формально ему принадлежит, но не используется (в дальнейшем из этого выросла технология ATM). Схема выглядит следующим образом: кроме того, что передаём данные в строках кадра, в начале каждой строки добавляем небольшой заголовок, который несёт информацию о классе или типе передаваемого контента. Фактически внутри ширины пропускной способности, выделяемой каждому абоненту, создаем ещё собственные виртуальные каналы, которые называются ***виртуальными путями***. Виртуальный канал – фактически идентификатор абонента (как IP или MAC-адрес), а виртуальный путь – идентификатор службы, использующей трафик (как TCP порт). Таким образом, получается размытие синхронной схемы – если отходим от привязки только к абонентам и привязываемся к конкретным службам, то почему нельзя использовать пустующую пропускную способность других абонентов? Это и положено в основу протокола АТМ.

В классической схеме синхронных иерархий с виртуальными каналами нам неважно, какой трафик мы передаём в каждом виртуальном канале. Об этом заботится арендатор канала, который загружает трафик в канал и потом на обратной стороне его извлекает. Задача мультиплексирующего оборудования – только направить этот трафик на конкретный железный ящик, который этот трафик преобразует во что-то более пригодное к употреблению - либо IP-маршрутизатор, либо гейт в телефонную сеть, либо гейт в кабельную телевизионную сеть и т.п. А если начинаем работать с внутренним мультиплексированием и переходим от идеологии виртуальных каналов к идеологии виртуальных путей, то нам требуется более детальное понимание, что делать с этим трафиком. То есть после этого мультиплексоры из тупых устройств (основная задача которых – только скомпоновать трафик, навесить служебные заголовки и передать в линию), превращаются в устройства достаточно интеллектуальные, которые должны по идентификатору виртуальных путей определять именно то оборудование, которому этот трафик предназначен.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Level** | **Америка** | | | **Япония** | | | **Европа** | | |
| **Проп. спос.** | **Коэф. мульт.** | **Кванты** | **Проп. спос.** | **Коэф. мульт.** | **Кванты** | **Проп. спос.** | **Коэф. мульт.** | **Кванты** |
| 0 | 64 | – | 1 | 64 | – | 1 | 64 | – | 1 |
| 1 | 1544 | 24 | 24 | 1544 | 24 | 24 | 2048 | 32 | 32 |
| 2 | 6312 | 4 | 96 | 6312 | 4 | 96 | 8448 | 4 | 128 |
| 3 | 44736 | 7 | 672 | 32064 | 5 | 480 | 34368 | 4 | 512 |
| 4 | 274176 | 6 | 4032 | 97728 | 3 | 1440 | 139264 | 4 | 2048 |
| 5 | – | – | – | – | – | – | 564992 | 4 | 8192 |

Значения пропускной способности в различных стандартах синхронных цифровых иерархий.

* *Пропускная способность* в килобитах
* *Коэффициент мультиплексирования* – сколько каналов предыдущего уровня иерархия вкладывается в текущий
* *Кванты* - сколько каналов наименьшего уровня вкладывается в текущий

На 0 уровне – частные компании, 1 – небольшие провайдеры, 2 – провайдеры уровня крупных городов, 3 – провайдеры уровня областей, 4 – глобальные телекоммуникационные компании. На втором уровне американский и японский стандарты совпадают, потому что ранние уровни иерархии в Японии делались американцами. После этого стандарты расходятся - связано это с тем, что Япония – страна поменьше, и там было меньше телекоммуникационных компаний, поэтому коэффициенты мультиплексирования меньше. В США 6 компаний, занимающихся междугородной телефонной связью, крупнейшая из них - АТ&Т. Европейский стандарт развивался с отставанием, поэтому имели возможность посмотреть на слабые места конкурентов, что-то исправить и сделать более логично.

Поговорим теперь об ***асинхронных схемах***. Примером является ***протокол-технология ATM*** (Asynchronous Transfer Mode, появилась в 80-е годы). Идеи появились раньше, в первую очередь идея внутреннего мультиплексирования. Первый стандарт появился в 1988 году. Особенность: всё пространство каналов связи рассматривается как плоское, т.е. мультиплексирование выступает как внутренняя техническая функция и не отражается на свойствах данных. Как мультиплексируются каналы – не играет роли. Протокол может использоваться и в локальных сетях без мультиплексирования, и такие даже были, но массовостью не пользовались. В технологии используются фреймы длиной 53 октета, которые называются ***ячейки ATM***. Это, видимо, мировой рекорд на минимальную длину фрейма.

Откуда взялся такой размер? Чем меньше длина фрейма, тем меньше реальная потеря пропускной способности при мультиплексировании трафика. Если сравнить с фреймом STM-1 – у него размер полезной нагрузки 261 октет на виртуальный канал на 1 фрейм, т.е. если занимаем один виртуальный канал, то, сколько бы данных ни передавали, всё равно займём 261 октет. То есть возникает существенный оверхед на передачу коротких данных. А кодеки, используемые в цифровой телефонии, тяготеют к использованию небольших порций данных размером всего несколько октетов. Например, кодек GSM передает данные с минимальным размером полезной нагрузки без GSM-заголовка – 28 октетов. Представьте, что вы будете маршрутизировать GSM-трафик через STM. Если решать задачу технически, то трафик будет уплотняться, т.е. GSM-фреймы будут упаковываться в блоки меньшей длины, но нас интересует теоретический вопрос. Тогда 261-28 = 233 октета пропадают впустую - 90% трафика теряется. Поэтому задача в том, чтобы использовать как можно более короткие ячейки. В Европе, где в глобальных сетях в основном существовал ещё телефонный трафик, настаивали на размере полезной нагрузки ячейки в 32 октета. Это близко к 28 октетам GSM, потому что кодеки цифровой телефонии все похожи. В США, где уже набирал силу поток компьютерных данных, наоборот, хотели ячейки большей длины - иначе большой оверхед на передачу собственного заголовка. Поэтому американцы настаивали на 64 октетах. В результате международный союз по телефонии (ITO – International Telephony Union) решил – полезная нагрузка будет ***48 октетов***. При таких спорах это помогает. 5 октетов – по-видимому, тоже рекорд на минимальный размер заголовка.

В результате того, что используются значения, близкие к предельным, ***существует два типа ATM-трафика*** и, соответственно, ATM-ячеек – для пользователей ATM и для трафика между двумя ATM-маршрутизаторами. Как мы уже говорили про синхронную иерархию с мультиплексированием, вводятся два понятия – виртуальный канал (идентификатор абонента, аналог TCP-порта) и виртуальный путь (идентификатор сервиса, который предоставляется конечному пользователю, аналог IP-адреса).

В 80-е годы в Советском Союзе разрабатывали свой собственный протокол для телефонии на основе асинхронного мультиплексирования, и в начале 90-х годов многие советские инженеры, которые разрабатывали проект, либо уехали на запад, либо активно с западом сотрудничали, и многие их идеи влились в современную реализацию ATM. Предполагалось использовать тот же самый подход, что и в ATM, но 1) с трёхуровневой адресацией (в ATM двухуровневая – виртуальный канал и виртуальный путь), 2) предлагалось использовать ячейки разной длины. Это решило бы многие проблемы ATM. Почему в ATM отказались от разной длины ячеек? Чтобы снизить реалтайм-нагрузку на ATM-коммутаторы, потому что в случае разной длины ячеек должно быть поле, ответственное за длину ячейки/полезной нагрузки, и на обработку такой информации должно тратиться дополнительное время. А одной из целей при изначальной выработке стандарта ATM (1983 г.) была возможность реалтайм-обработки трафика без создания заторов на коммутаторах на том оборудовании, которое тогда было доступно. Обработка ячеек переменной длины, видимо, тогда была невозможной. Неизвестно, как хотели справиться с этим в СССР. Посмотрите на верхние уровни иерархий: 274 Мбит, 564 Мбит. На тот момент надо было постараться сделать такое оборудование, которое могло осуществлять коммутации на таких скоростях.

***Ячейка типа пользователь-сеть (user-network)***

|  |  |  |
| --- | --- | --- |
| CFC | VPI | |
| VPI | VCI | |
| VCI | | |
| VCI | PT | CLP |
| ECRC8 | | |

Так как размер заголовка всего 5 октетов, то заполнен он очень плотно.

* *Common Flow Control* (CFC, 4 бита) – чем выше, тем больше вероятность, что эта ячейка может быть уничтожена коммутатором в случае перегрузки сети. Фактически, отчасти определяет и тип трафика. В ячейках для передачи компьютерных данных – более низкое значение, для телефонных данных – более высокое. Из-за чего в GSM и используют такие короткие ячейки - даже если потерять одну ячейку, абонент всё равно в состоянии понять собеседника.
* *VPI* (1 октет) – идентификатор виртуального пути (идентификатор службы, предоставляемой конкретным ATM-провайдером).
* *VCI* (2 октета) – идентификатор виртуального канала (адрес ATM-устройства). Вместе пару VPI-VCI называют маршрутным полем ячейки.
* *Payload Type* (тип полезной нагрузки, 3 бита) – набор флагов, которые определяют тип инкапсулируемого протокола – реалтайм-передача / нереалтайм с поддержкой целостности данных. Таким образом, на поведение маршрутизатора при уничтожении ячейки влияют два поля – CFC и PT. Также в этом поле есть флаг, определяющий, какого типа ячейка: user-network или network-network.
* *Cell Lost Priority* (приоритет потери ячейки, 1 бит) – может ли вообще быть уничтожена эта ячейка. В ATM считается, что любой пользовательский трафик может быть уничтожен. Защищён этим полем может быть только служебный трафик между устройствами ATM. Если в этом поле 1 – ячейка может быть отброшена, если 0 – несмотря на значение CFC и PT, эта ячейка неуничтожаема.
* *Enhanced CRC8* (контрольная сумма, 1 октет). В ранних версиях ATM это поле не использовалось, и для совместимости с оборудованием первого поколения приходилось на уровне обработчика протокола игнорировать поле контрольной суммы. Причина – фирма Motorola, один из крупнейших игроков на рынке ATM-маршрутизаторов, выпустила первый маршрутизатор ещё до выработки официального стандарта. Потом в стандарте решили перейти от использования CRC8 к ECRC8, потому что он вроде бы быстрее считается. В результате того, что в Мотороле всё это было аппаратно зашито, им пришлось отказаться от вычисления контрольной суммы вообще.

***Ячейка типа сеть-сеть (network-network)***

|  |  |  |
| --- | --- | --- |
| VPI | | |
| VPI | VCI | |
| VCI | | |
| VCI | PT | CLP |
| ECRC8 | | |

Нет поля CFC, расширено поле VPI – и соответственно расширен набор служб, которые используются для обслуживания сети ATM. Такие ячейки используются для передачи служебного управляющего трафика ATM-маршрутизаторов. ***Весь трафик служебный***, поэтому CFC не нужно – либо однозначно надо сохранять ячейку, и тогда за это отвечает поле CLP, либо (если в CLP 1) маршрутизатор сам решает, уничтожить ли эту ячейку, не глядя на вероятности в CFC. Но в основном трафик network-network защищён ячейкой CLP. В ATM с самого начала закладывались средства автоматического мониторинга и управления сетью. Если используется ATM-сеть на коммутаторах, сделанных по стандарту, то с очень большой вероятностью в тонкую настройку сети влезать не надо, настраивается сама в соответствии с текущими параметрами – текущая пропускная способность, средний процент ошибки на октет, количество общих сбоев сети. Всё, что требуется – назначить адреса VCI, VPI и, если требуется, переопределить значение поля CFC для конкретных видов трафика. Остальные данные жёстко завязаны на используемый инкапсулируемый протокол и используемое оборудование.

Протокол ATM обеспечивал близкую к предельной пропускную способность, и ***оборудование было дорогим***. Сетевая плата ATM для компьютеров в начале 2000-х годов стоила 3500 долларов. Стоимость ATM-коммутатора пользовательского уровня (аналог – Ethernet-свитч) в минимальной конфигурации с поддержкой 4 виртуальных каналов – 12-15 тысяч долларов. 30 каналов – 50 тысяч долларов. Пик технологии ATM пришёлся на 90-е годы. Когда стало ясно, что глобальный трафик непрерывно растёт и конца не видно, а уже созданные синхронные сети с мультиплексированием надо использовать (переделывать дорого), то для таких сетей ATM стал очень хорошим решением. Позволял выжать из этих сетей по максимуму - оверхед порядка 10%. Но оборудование ATM в силу своей развитой интеллектуальности дешеветь не собиралось. И в 2000-е годы появилась технология, которая убила ATM – гигабитный оптический Ethernet. Проще было сделать канал с небольшим запасом по пропускной способности и использовать гигабитный/10-гигабитный Ethernet, чем покупать дорогое магистральное оборудование для АТМ.

Различные технологии внутри ATM. Внутри ATM можно построить такую же иерархию стека протоколов, который называется ***ATM Adaptation Layer***. ATM от всех других технологий отличается тем, что реально в ATM не инкапсулируются протоколы более высокого уровня. Трафик более высокого уровня демультиплексируется – нарезается кусками по 48 октетов и запихивается в ячейки ATM, т.е. ATM-трафик представляет огрызки того, что в него насовано. В связи с этим важная задача, которую решает ATM – передача ячеек строго в том порядке, в котором они отправлены. Ни одна ячейка не может обогнать другую, иначе всё демультиплексирование идёт лесом.

ATM определяет ***пять классов служб*** – они различаются по среднему битрейту, по необходимости организовывать передачи данных в реальном времени, по необходимости установления соединения между отправителем и получателем данных. Adaptation Layer – некоторая более ранняя классификация, чем A, B, C, D, X.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Class** | **A** | **B** | **C** | **D** | **X** |
| **Bitrate** | CBR | VBR | VBR | ABR | UBR |
| **Real-time Transfer** | + | + | + | – | – |
| **Connection** | + | + | + | +/– | – |
| **Adaptation Layer** | 1 | 2 | ¾ | ¾ | – |
| **Traffic** | E1,T1 (эмуляция синхронной) | Видео, мультимедиа | Цифровая телефония | Компьютерные данные | |

* *CBR (constant bitrate)* – передача с постоянной скоростью, введена для совместимости с синхронной цифровой иерархией и используется там, где надо сохранить логику работы синхронных мультиплексоров с используемым ATM-оборудованием.
* *VBR (variable bitrate)* – передача данных со скоростью, про которую мы ничего не можем сказать. Знаем только верхнюю предельную скорость передачи данных, а нижняя скорость может быть любой.
* *ABR (arbitrary bitrate)* – скорость данных ограничена, но на короткое время может превышать предел. В среднем данные передаются с постоянной скоростью, но иногда могут возникать локальные превышения.
* *UBR (user bitrate)* – вообще класс X был введён значительно позднее, ближе к 2000 году, с появлением сетей ISDN и ADSL, поверх которых пускался протокол ATM, и специально предназначался для описания пользовательского трафика. То есть это трафик, где нельзя сделать никаких предположений о скорости передачи.
* *Real-time Transfer* – необходима ли передача в реальном времени без задержек. Это характерно только для классов A, B, C. Для них же характерно установление соединения средствами ATM.
* *Установление соединения средствами ATM*. В логике ATM установление соединения – это простая операция, когда перед отправкой данных (организацией виртуального канала) проверяется, во-первых, наличие данных, во-вторых, готовность к приёму, чтобы не гнать впустую данные по сети. В классе D – опциональная установление соединений. В классе X установление соединения не предусмотрено вовсе, потому что основной тип передаваемых данных – компьютерные данные, и скорее всего там поверх ATM идёт протокол TCP, который сам устанавливает соединение, и проводить дополнительную проверку на установление соединения просто не требуется.

Протокол Frame relay – тоже предназначен для передачи данных в глобальных сетях. Решение проблемы последней мили. Об этом пишет Миша Винник.