**Лекция за 14.03.2012, часть 2**

***Фрейм Ethernet DIX***

Компания Novell выпустила собственный продукт со своим типом фрейма (***802.3 Novell***). ***Novell NetWare*** могла бы завоевать весь рынок, и никому больше из участников комитета ничего бы не досталось, если бы они ничего не предприняли. Естественно, что у них были какие-то свои наработки и что они находились в условиях довольно жёсткого цейтнота (***цейтно́т*** — в настольных играх— недостаток времени для обдумывания хода). Хотя цейтнот понятие относительное: вся эта деятельность длилась годами, поэтому срок измерялся десятками месяцев.

На тот момент казалось, что в связи с политикой *Novell*'а комитет *802.3* вообще развалился, поэтому большая тройка *DEC*, *Intel* и *Xerox* выпустила своё собственное решение ***Ethernet II*** (***Ethernet DIX***). II — второе поколение по сравнению с ***Aethernet***, а не с ***802.3 Novell*** (который они вообще за *Ethernet* не признавали). Они создали фрейм, который вообще не соответствовал стандарту *MAC*.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DA** | **SA** | **Type** | **Payload** | **CRC32** |
| 6 | 6 | 2 | 46-1500 | 4 |

1. Ethernet DIX

С формальной точки зрения, этот фрейм в точности повторял novell'овский, однако третье поле заголовка получило название и назначение не длины, а «типа», где ***тип*** должен кодировать протоколы более высокого уровня, инкапсулирующиеся внутрь этого фрейма. Нарушается стандарт *MAC*, т. к. теряем информацию о длине фрейма. В своё время, это составляло очень серьёзную проблему: для настройки работы *Ethernet*'а с фреймом ***DIX*** системные администраторы должны были пройтись по всем хостам в сети и вручную выставить длину используемого фрейма или согласиться на работу с фреймом максимальной длины (***1518***). Если этого не сделать, то сеть нормально работать не сможет, т. к. никакого алгоритма автоматической селекции длины фрейма на тот момент не существовало (а локальных сетей у предприятия может быть несколько, в каждой по сотне хостов, поэтому это действительно было проблемой).

Но было и существенное преимущество по сравнению с novell'овским фреймом: ***DIX*** фрейм позволял инкапсулировать в себя любую полезную нагрузку. Активно использовался для инкапсуляции ***IP***, использовался в сетях ***AppleTalk*** (локальная вычислительная сеть) и других. Также включили официальную поддержку фреймов ***IPX*** в качестве полезной нагрузки фреймов ***DIX***.

***Историческая справка***

Это всё произошло примерно в *1992*. Таким образом, большой тройке потребовалось 3 года (2.5) работы на создание своего гениального решения (*1989* — демонстрация *Novell*'а и начало продаж ***Novell NetWare***). На самом деле это были три года плодотворной работы: были разработаны сетевые адаптеры, новое ПО. Большим подспорьем оказался выпуск компанией *IBM* своей ОС ***OS/2***, и по разным причинам *IBM* включила в ***OS/2*** поддержку именно фреймов ***DIX***, а не novell'овских (видимо, т. к. *IBM* на тот момент активно конкурировало с *Novell*'ом и посчитало, что будет полезнее поддержать аутсайдеров, чем *Novell*'а). Этим, возможно, *IBM* и вырыли себе яму, т. к. они делали упор на ***Token Ring***.

Эти два фрейма начали конкурировать между собой. Причём выигрывать стал ***DIX***, потому что он поддерживал хотя бы теоретическую возможность работы с фреймом ***IPX***, а novell'овский фрейм ни с чем другим работать не мог, т. е. при установке ***Novell NetWare*** о поддержке ***IP*** можно забыть (по понятным причинам). На самом деле теоретическая возможность сосуществования в сети фреймов различных типов есть, но подобная селекция требует довольно больших затрат процессорного времени, чего не было в то время. В конце концов, в ***Novell NetWare*** (с третьей версии) появилась поддержка как фреймов ***802.3 Novell***, так и фреймов ***DIX***. С этого момента novell'овский фрейм перестал упоминаться в анналах истории.

***Фрейм 802.2 Ethernet (Ethernet LLC)***

В это время в комнате *802.2* продолжала собираться рабочая группа по протоколам канального уровня, поэтому впоследствии к *1994* году появляется первый стандарт канального уровня ***LLC*** (***Logical Link Control***).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **DA** | **SA** | **Length** | **DSAP** | **SSAP** | **Ctrl** | **Payload** | **CRC32** |
| 6 | 6 | 2 | 1 | 1 | 1(2) | 42-1497 | 4 |
| *MAC* | | | *LLC* | | |  |  |

2. Ethernet LLC

***DSAP*** = ***Destination Service Access Point***

***SSAP*** = ***Source Service Access Point***

Заголовок ***LLC*** включал 3 поля общей длиной 3 (4) октета. Первые два поля ***DSAP*** и ***SSAP*** предназначались для селекции стека протоколов более высокого уровня, который должен был осуществлять обработку полезной нагрузки соответствующего фрейма.

Важный момент: середина 90-х, интернет только-только появляется, большинству участников кажется, что будущий интернет будет чрезвычайно гетерогенным (в том, что он будет, никто не сомневался), что он будет включать сети разные как по физическому уровню, так и по сетевому и транспортному. В частности, предполагалось, что одни и те же протоколы могут быть реализованы в десятках видов различных стеков. Искались универсальные решения, ***LLC*** с этой точки зрения оказался довольно удачным.

Третье поле заголовка (переменной длины, что сделало много проблем для программистов) — поле ***Control*** («*управляющее поле*») содержало флаги тонкой настройки. По значению первого бита этого поля можно было сразу сказать, было ли оно одно или двух-битным. 0 — короткий контрол из одного октета; 1 — длинный контрол из двух октетов. В длинном контроле могли присутствовать опциональные флаги.

В поле ***Control*** были флаги проверки целостности данных на физическом уровне и подтверждение доставки на физическом уровне. Сейчас и то, и другое эффективно реализуется на уровне ***TCP***, и на физический уровень никто с этим не лезет, хотя существовали протоколы, которые активно использовали эти возможности формата ***LLC*** (например, ***NetBEUI*** от *Microsoft*). Подробную информацию о флагах можно брать из английской википедии.

Использование этого заголовка сократило размер полезной нагрузки до ***42-1497*** октетов, а ***CRC32*** остался там же. Фрейм получил название ***802.2 Ethernet*** или ***Ethernet LLC***. Это первый стандарт *Ethernet*, который начала поддерживать компания *Microsoft*. На этот же фрейм перешла и компания *Apple*.

***Противоречивость LLC***

Тогда же выяснилось, что одного октета для кодирования даже не всего стека протоколов более высокого уровня, а одного протокола, не хватает. Связано это с тем, что в *Ethernet* инкапсулировали всё подряд: ***IP***, ***IPX***, ***AppleTalk*** и т. п., а также с тем, что разные производители ПО и оборудования начали присваивать одним и тем же протоколам нумерацию как придётся. В результате чего пространство имён начало очень резко исчерпываться. В этом и заключается внутренняя противоречивость пространства ***LLC***, т. е., с одной стороны, предусматриваем его на все случаи жизни, даже оставляя возможность кодирования на уровне отправителя и получателя в разных стеках протоколов, а с другой — оставляем очень узкое пространство имён в размере одного октета.

Это составило довольно серьёзную проблему, более того ***IEEE*** начали всех производителей ПО и оборудования призывать переходить на формат ***LLC*** (отчасти из-за быстрого исчерпания пространства имён). Так что *всех* ***LLC*** не удовлетворил.

***Фрейм Ethernet SNAP***

Последняя итерация в этой цепочке привела к созданию следующего типа фрейма, получившего название ***Ethernet SNAP*** (***Simple Network Access Point***, *упрощённая сетевая точка доступа*). Упрощённая по сравнению с той, которая была в стандарте ***LLC***, включавшая раздельные точки доступа для отправителя и получателя. Поскольку было острое желание сохранить все разработанные до этого стандарты, то этот тип фрейма инкапсулировался внутрь фрейма ***802.2***.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | 0xAA | 0xAA | 0x03 | 0x00000000 |  |  |  |
| **DA** | **SA** | **Length** | **DSAP** | **SSAP** | **Ctrl** | **OUI** | **Proto** | **Payload** | **CRC32** |
| 6 | 6 | 2 | 1 | 1 | 1 | 3 | 2 | 37-1492 | 4 |
| *MAC* | | | *LLC* | | | *SNAP* | | | |

3. Ethernet SNAP

***OUI*** (***Organisation Unit Identifier***) — организация, которая присваивает номера протоколов. Трёх-октетное, т. к. предполагалось, что будет очень много организаций, присваивающих номера протоколам.

Двух-октетное поле ***Protocol***, отвечающее за номер протокола, который инкапсулировался внутрь полезной нагрузки. Была и альтернативная позиция: формально сохранив ***LLC*** заголовок, сделать вместо двух полей DSAP и SSAP одно общее поле, нумерующее инкапсулируемый протокол. Но в итоге получился самый тяжеловесный из имеющихся *Ethernet* фреймов.

Заголовок ***LLC*** оказался не нужен и превратился в «*мертвый заголовок*», т. е. его поля имеют фиксированные значения, приведённые выше (в гексе). ***DSAP***==***SSAP***==***0xAA***, т. к. они на тот момент ещё не были никому присвоены; ***Control***==***0x03***, т. е. протокол не поддерживает на физическом уровне ни целостность данных, ни контроль доставки.

В настоящее время поле ***OUI*** тоже имеет фиксированное значение, а именно нулевое, это идентификатор организации ***ISO*** (***International Organization for Standardization***). Таким образом, 3/5 заголовка ***SNAP*** тоже оказались не нужны.

***Историческая справка***

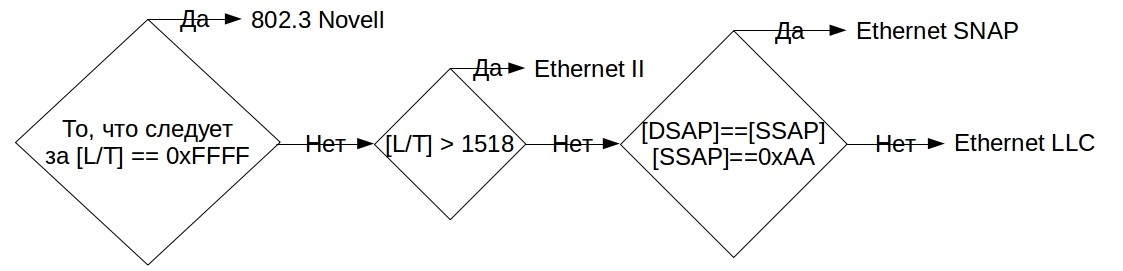
Все эти события произошли во второй половине *90*-х: ***Ethernet DIX*** *1992* — начало *1993* года, ***Ethernet 802.2*** *1994*, ***Ethernet SNAP*** конец *1995* — начало *1996*. На тот момент уже появились достаточно мощные компьютеры (первые *Pentium*'ы), и стало возможным проводить селекцию фреймов на лету. Значит, казалось бы, в одной сети могли присутствовать фреймы всех 4-х типов, и гетерогенность сетей подтвердилась.

Но тут на сцену вышел *Microsoft*. *95*-*96*-й год — это выход ***Windows 95***. Параллельно с ней разрабатывалась и серверная ОС ***Windows NT***, поддерживающая все эти четыре фрейма. А вот ***Win95*** поддерживал только фрейм ***Ethernet DIX*** (что крайне удивило всех). Возможно, связано с складывающимся альянсом *Intel*'а и *Microsoft*'а, и *Intel* это решение пролоббировал. А *Intel* на тот момент был производителем (вместе с *DEC*) примерно трети чипов для сетевых карт.

Ситуация с ***Win95*** решила все дела, т. к. она действительно стала массовой (т. е. компьютеры на её основе стали активно объединяться в сети). В результате, выжил именно ***Ethernet DIX***. На данный момент, именно ***Ethernet DIX*** является стандартом де факто для *Ethernet*'а.

***WinXP***, ***WinVista***, ***Win7*** поддерживают всё, т. к. ***WinNT*** поддерживала всё, но реально остальные фреймы больше нигде сейчас не используются. Дольше всех продержался *Apple*, т. к. старался честно соблюдать стандарты. Он использовал в качестве основного фрейма ***Ethernet SNAP*** до перехода на *Intel*'овскую платформу.

***Селекция фреймов Ethernet***



Первая проверка заключается в том, что проверяем, не начинается ли то, что следует за полем ***Lenght***/***Type***, с октетов ***0xFFFF***, потому что это стандартная преамбула заголовка ***IPX***, т. е. ***802.3 Novell***. Потом проверяем значение поля ***Length***/***Type***. Максимальная длина фрейма ***1518*** октетов, т. е. если это значение больше, чем ***1518***, то это поле кодирует не длину, а тип данных, тогда это поле ***Ethernet II*** (изначально нумерация протоколов в ***DIX***'е шла с чисел больших ***1518*** для того, чтобы этот фрейм нельзя было перепутать с novell'овским). Далее проверяем, если ***DSAP*** равно ***SSAP*** и равно ***0xAA***, то имеем дело с ***Ethernet SNAP***, иначе — ***Ethernet LLC***. Таким образом, проверка происходит в три этапа.

Реализовать её можно было с самого начала, но, к сожалению, в течение долгого времени было довольно жёсткое ограничение на вычислительные способности, и такая проверка сильно затормозила бы обработку фреймов.

***Физический уровень сети***

Вроде бы, фрейм ***DIX*** не поддерживает работу с фреймами переменной длины. На самом деле, это довольно существенная проблема (она была описана ранее). Сейчас эта проблема решена и решена довольно интересным образом. Середина *90*-х — начало появления сетей на основе витой пары. При работе поверх витой пары меняется логика работы сети. При работе поверх коаксиального кабеля, когда передачи не было, никакого сигнала в линии не было. В настоящее время, в линии всегда присутствует сигнал заполнитель, и каждый *Ethernet*-фрейм, передаваемый в сеть, имеет перед собой так называемую преамбулу. Это 8 октетов, имеющих фиксированные значения, которые никогда не могут быть частью данных при том механизме кодирования данных, который используется в *Ethernet*'е.

В настоящее время, физический уровень любой сети (стека ***ISO OSI*** в частности) принято разделять ещё надвое. То, что было раньше, относилось к ***Media Independent Interface***, а то, что будет далее — глубокий физический уровень, а именно уровень представления данных непосредственно в канале передачи данных.

Как всё это выглядит непосредственно на физическом уровне?

|  |  |  |
| --- | --- | --- |
| **Preamble** | **Ethernet DIX** | **Interframe Gap** |
| 8 | 64-1518 | 12 |

4. Физический уровень сети

Каждый фрейм предваряется восьми-октетной преамбулой, после него следует сам фрейм, завершается всё это двенадцати-октетным ***Interframe Gap***'ом. И преамбула, и ***Interframe Gap*** имеют фиксированные значения. 7 октетов вида ***0x10101010***, 1 октет вида ***0x10101011***. Механизм шифрования в *Ethernet*'е таков, что гарантируется, что такая последовательность из восьми октетов никогда не встретится среди октетов данных. Поля самого фрейма сознательно не пишутся, чтобы показать, что мы работаем на более низком уровне стека. т. к. на самом деле, конечно, можно использовать в заголовке ***MAC***-адреса, которые начинаются с таких значений, но в результате кодирования данных получатся другие значения, не совпадающие с восьмёркой.

***Расширение 802.1Q (VLAN tag)***

Есть ещё одно дополнение, вводящееся в ***DIX***-фрейм. Это поле ***802.1Q*** (или ***VLAN tag***).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DA** | **SA** | **Type** | **VLAN tag** | **Payload** | **CRC32** |
| 6 | 6 | 2 | 4 | 42-1496 | 4 |

5. Расширение 802.1Q

Непосредственно за полем типа идёт поле тэга. Оно даёт возможность создания на одном коммутаторе *Ethernet* нескольких логических непересекающихся между собой сетей, информация которых друг в друга передаваться не будет. Таким образом, на одном физическом коммутаторе можно создать несколько логических.

*Итог*: в настоящее время повсеместно используется фрейм ***DIX*** (возможно с расширением ***802.1Q***), фрейм может иметь переменную длину, т. к. информация о начале и конце фрейма в данный момент содержится не в самом фрейме (в заголовке), а извлекается из использования преамбулы и ***Interframe Gap***'а, которые ограничивают с двух сторон этот фрейм. Этот фрейм используется в сетях как 100-мбитных, так и гигабитных, так и в сетях с более высокой пропускной способностью. Также он используется в беспроводных сетях ***802.11***.

***Беспроводные сети***

В основном будет рассмотрен стандарт ***802.11*** (или, как он назывался до принятия официального стандарта, ***Radio Ethernet***). Отличительная особенность: использование ethernet'овского фрейма (***DIX***). Таким образом эти сети могут прозрачно сопрягаться со стандартными ethernet'овскими сетями (например, точки доступа ***Wi-Fi***, совмещённые с ethernet-коммутаторами).

Отличие фреймов *Ethernet* и фреймов ***Wi-Fi***: в сетях ***Wi-Fi*** используются сами по себе фреймы, внутрь которых инкапсулированы фреймы *Ethernet*. Фактически, в качестве полезной нагрузки физического уровня ***Wi-Fi*** используются фреймы физического уровня *Ethernet*. Это приводит к некоторой избыточности по передаче служебной информации, но компенсируется тем, что возможна прозрачная трансляция из одной сети в другую.

С точки зрения пользователя есть набор стандартов: ***802.11*** (***a***), ***b***, ***g***, ***n***. В России применение оборудования ***802.11 a*** запрещено, т. к. он использует радио диапазон, который не предоставлен государственной комиссией по радио частотам в свободный доступ (его используют различные спецслужбы).

Все системы беспроводной связи используют диапазон частот в районе нескольких гигагерц. Так получилось, что этот диапазон практически во всех странах в своё время давался спецслужбам. Поэтому иногда возникают проблемы при использовании различного беспроводного оборудования.

Эти стандарты отличаются скоростью: ***a*** — до ***34*** мбит/с, ***b*** — ***11*** мбит/с, ***g*** — ***54*** мбит/с, ***n*** — либо ***150***, либо ***300*** мбит/с. Существовали различные частные расширения стандартов, например, ***g+*** (разработан при участии компании *Asus*), обеспечивающий скорость до ***100*** мбит/с при условии, что все устройства сети поддерживают ***g+***.

***Принципы построения беспроводной сети***

При подключении новой точки доступа нужно сначала выбрать диапазон, в котором она работает (стандарты до ***g*** включительно поддерживают ***11*** диапазонов, стандарт ***n*** поддерживает ***15*** диапазонов). Это позволяет развести по разным поддиапазонам близко расположенные точки доступа. Современные беспроводные устройства имеют автоматическую настройку и в большинстве случаев решают эту проблему сами. В ранних устройствах надо было самим перебирать доступные поддиапазоны.

Топология сетей ***Wi-Fi*** является топологией ***общей шины*** с *разделением во времени*, т. е. все устройства, подключённые в данный момент к ***Wi-Fi***, в течение всего срока работы синхронизируют свои часы (высокоточные таймеры), и в отведённые им промежутки времени у них разрешена передача. Это в большинстве случаев избавляет от проблемы коллизий (за исключением малого процента, связанных с некорректной настройкой или ошибками в ПО тех или иных устройств). *Исключение*: ранний стандарт ***b*** не имел временного разделения, там была обыкновенная общая шина, с которой и связана низкая скорость ***11*** мбит/с.

Большинство современных точек доступа умеют работать с несколькими стандартами или, по крайней мере, поддерживать режимы совместимости. В режимах совместимости нет гарантии того, что будет достижима максимальная скорость передачи данных, особенно при совместимости с устройствами типа ***b*** (наличие устройства типа ***b*** в сети, то сеть практически сразу просядет).

***Безопасность беспроводных устройств***

Существует несколько протоколов шифрования, обеспечивающих безопасную передачу данных по ***Wi-Fi***. Самый простой вариант — вообще без шифрования (так работают публичные точки доступа в ресторанах и т. п.). Первый протокол шифрования — ***WEP*** (***Wireless Enhanced Protection***), включавший в себя симметричное шифрование с ключом длиной ***64***, ***128*** или ***256*** бит. Современное оборудование этот ключ ломает за разумное ограниченное время. Поэтому в реальных сетях он сейчас не применяется.

Следующее поколение по безопасности — это протокол ***WPA*** (***Wireless Protection Architecture***), который имеет 2 поколения (***WPA*** и ***WPA2***), отличающиеся поддерживаемым набором алгоритмов шифрования. В отличие от ***WEP***, ***WPA*** поддерживает шифрование с открытым ключом. На самом деле это только начальный ключ (в *Microsoft* его называют пароль), используемый при входе в беспроводную сеть. Он используется один раз в одном сеансе для шифрования сессионных ключей. Таким образом, в технологии ***WPA*** используются ключи на каждую сессию, которые обновляются примерно раз в полчаса (можно настроить по-другому). Обычно за это время с использованием современного оборудования дешифровать ***WPA***-ключ не представляется возможным (если не использовать, например, суперкомпьютер «*Ломоносов*» из *МГУ*).

Ещё одним важным достоинством протокола ***WPA*** является то, что с его помощью можно и хранить пароль непосредственно на точке доступа, и использовать системы централизованной аутентификации. В этом случае нужно указывать в настройках точки доступа провайдера аутентификации. Именно он и занимается назначением сессионных ключей. Это удобно в централизованных системах больших предприятий. Один раз войдя в сеть устройства, пользователь сможет подключаться и к его локальной сети, и к его ***bluetooth***'у, используя разные ключи, но их смена будет происходить прозрачно для пользователя.

Первое поколение протокола ***WPA*** использовало схему шифрования ***TKIP***, ***WPA2*** использует шифрование ***AES*** (некое развитие протоколов шифрования ***DES***).

***Wi-Fi*** может быть в варианте ***точка-точка***, когда 2 устройства связываются по ***Wi-Fi*** каналу между собой, и в режиме ***infrastructure*** (*инфраструктурный режим*), когда есть одна точка доступа, к которой подключаются несколько беспроводных устройств. При этом между собой эти устройства обмениваются данными только через точку доступа.

В настоящее время есть идея, в которой предполагается, что беспроводные устройства смогут общаться друг с другом непосредственно, передавая данные друг другу, а не через точку доступа, точка доступа будет выступать только в качестве координатора сеансов и будет служить для аутентификации устройств пользователей и обеспечения безопасности. Если эта идея будет реализована, то будет возможно заметное увеличение пропускной способности беспроводных сетей, особенно с разнородными устройствами, активно взаимодействующих друг с другом. Т.е. это решение предназначено для корпоративных сетей, у которых есть заметный внутренний трафик данных.

Сейчас беспроводные сети особенно в Штатах, в меньшей степени в Европе и ещё в меньшей степени в России начинают использоваться именно в корпоративном секторе, в связи с резким сокращением затрат на инфраструктуру кабельной сети.