Глава 4. Адресация в многоуровневой модели

В данной главе будет рассмотрено:

- Различные системы счисления;
- Структура MAC-адреса, процесс формирования группового MAC-адреса и широковещательного MAC-адреса;
- Двоичная и шестнадцатеричная системы счисления. Методы перевода чисел из различных систем счисления;
- Работа ARP-протокола;
- Пример пересылки данных по сети.

4.1. Системы счисления

Перед тем, как мы начнем изучать системы счисления, отличные от привычной нам десятичной, хотелось бы напомнить, что электронные устройства воспринимают все возможные числа и связанные с ними сущности: адреса, номера портов, значения параметров протоколов и т.д. исключительно как последовательность нулей и единиц. Узлы и промежуточные устройства общаются друг с другом посредством нулей и единиц. Для более простого понимания процесса адресации мы используем десятичную систему счисления, в то время как машина изменяет ее в двоичную.

4.1.1. Двоичная система счисления

Двоичная система счисления является позиционной системой счисления. Это означает, что значение цифры в числе зависит от ее позиции. Количество используемых цифр называется основанием системы счисления. Основанием двоичной системы счисления является 2, так как данная система счисления состоит всего из двух цифр 0 и 1, называемых битами (от англ. *bit*, сокращение от ***bi**nary digi**t*** – двоичная цифра).

При переводе числа из десятичной системы счисления в двоичную число записывается как сумма степеней двойки. В двоичном представлении единицы находятся в разрядах, где есть соответствующие степени двойки, в остальных разрядах стоят нули.

Вспомним степени двойки: $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $2^6 = 64$, $2^7 = 128$, $2^8 = 256$.

Для удобства распределим их по разрядам. Восьми разрядов будет достаточно.

Таблица 4.1.1.1. – Десятичное представление чисел с разными степенями двойки

2^x	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Десятичное	128	64	32	16	8	4	2	1

Переведем число 13 в двоичную систему счисления. Представим, какие варианты со степенями числа 2 можно в этом случае использовать:

$$13 = 8 + 4 + 1 = 1x2^3 \times 1+2^2 \times 0x2^1 \times 1x2^0$$

Возможно ли представить число 13 как 3×4 + 1? Нет, так как система двоичная, и количество чисел в разряде может быть либо 0 (его нет), либо 1 (оно есть). Воспользуемся вышеприведенной таблицей в качестве шпаргалки.

Таблица 4.1.1.2. – Двоичное представление десятичного числа 13

2^x	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Десятичное	128	64	32	16	8	4	2	1
Двоичное	0	0	0	0	1	1	0	1

Первые нули можно опустить, получаем, что десятичное число 13 в двоичной системе равно 1101. На языке математики это выглядит так:

$$13_{10} = 1101_2$$

Усложним задание: переведем в двоичный вид число 196. Его можно представить как: 128 + 64 + 4.

Таблица 4.1.1.3. – Двоичное представление десятичного числа 196

2^x	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Десятичное	128	64	32	16	8	4	2	1
Двоичное	1	1	0	0	0	1	0	0

$$196_{10} = 11000100_2$$

Для того чтобы из двоичного числа получить десятичное, нужно произвести обратную операцию. Преобразуем, например, двоичное число 10101000_2.

Таблица 4.1.1.4. – Преобразование двоичного числа 10101000 в десятичную систему счисления

2^x	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Десятичное	128	64	32	16	8	4	2	1
Двоичное	1	0	1	0	1	0	0	0

$$1\times128 + 0\times64 + 1\times32 + 0\times16 + 1\times8 + 0\times4 + 0\times2 + 0\times1 = \\ = 128 + 32 + 8 = 168$$

Навыков работы с восемью разрядами двоичных чисел будет вполне достаточно для понимания адресации протокола IPv4, так как IP-адрес состоит из 32 битов и для удобства чтения он разделен на 4 группы по 8 битов, каждая из которых называется *октет* (от лат. *octo* – восемь). Таким образом, 32х битный IP адрес состоит из 4 октетов.

То есть электронное устройство получает набор битов на сетевом уровне, который для него выглядит так:

$$11000000101010000000000000000001$$

Разделим этот сплошной набор нулей и единиц на октеты:

$$11000000.10101000.00000000.00000001$$

Затем переведем каждый октет в десятичное число:

$$192.168.0.1$$

Правда же, так гораздо удобнее, чем запоминать последовательность из 32 нулей и единиц?

Ранее мы уже упоминали MAC-адрес, который используется в целях адресации. Он представляет собой точно такую же последовательность нулей и единиц, но для человека удобной формой записи будет не двоичная, а шестнадцатеричная система исчисления. В ней нам дальше и нужно разобраться.

4.1.2. Шестнадцатеричная система счисления

Шестнадцатеричная система счисления, так же, как и уже знакомые нам двоичная и десятичная, является позиционной, то есть позиция цифры в разряде определяет значение числа, обозначаемого этой цифрой. Таким образом , мы считаем не группами по 10, а группами по 16 – так называемыми *гекстетами*.

Вспомним, как меняются разряды в десятичной системе счисления. Первый разряд имеет значения от 0 до 9. Если мы прибавим единицу к 9, то получим 10, то есть второй разряд (10^1) увеличится на единицу, а первый разряд (10^0) снова примет значение 0.

В шестнадцатеричной системе есть особенность: 9 + 1 ≠ 10, так как это бы означало, что произошла смена разряда на более высокий (16^1). То есть 10 в шестнадцатеричной системе означает 16 в привычной нам десятичной системе счисления. Как быть? Математики решили в качестве шестнадцатеричных цифр использовать буквы латинского алфавита: A, B, C, D, E и F.

Таблица 4.1.2.1. – Представление чисел в различных системах счисления

Десятичная система счисления	Двоичная система счисления	Шестнадцатеричная система счисления
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10

До 9 все операции сложения и вычитания в десятичной и шестнадцатеричной системах совпадают, например: 7 + 2 = 9. Не совпадают же следующие операции:

9 + 1 = A, A + 1 = B, B + 1 = C, C + 1 = D, D + 1 = E, E + 1 = F.

А вот F + 1 = 10_{16} = 16_{10}.

10_{16} + 10_{16} = 20_{16}, но для нас (в десятичной системе) это равно 32_{10}. И т.д.

Переведем в шестнадцатеричную систему число 250.

Пользуясь навыками перевода числа из десятичной системы счисления в двоичную, здесь потребуется разложить заданное число как сумму произведений степеней с основанием 2.

Таблица 4.1.2.2. – Двоичное представление десятого числа 250

2^x	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Десятичное	128	64	32	16	8	4	2	1
Двоичное	1	1	1	1	1	0	1	0

$$1\times128 + 1\times64 + 1\times32 + 1\times16 + 1\times8 + 0\times4 + 1\times2 + 0\times1 = \\ = 128 + 64 + 32 + 16 + 8 + 2 = 250$$

После получения двоичного числа $250_{10} = 11111010_2$ разделим его на тетрады (на каждые 4 разряда).

$$1111\ 1010$$

После разделения на тетрады обратимся к таблице 4.1.2.1 (нужный нам фрагмент этой таблицы представлен на рисунке 4.1.2.1) и увидим, что каждой тетраде из двоичной системы счисления соответствует свое число из шестнадцатеричной системы. Так, тетрада $1111_2 = F_{16}$, а тетрада $1010_2 = A_{16}$. Таким образом, число $250_{10} = 1111\ 1010_2 = FA_{16}$.

10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Рисунок 4.1.2.1. – Фрагмент таблицы с различными системами счисления

Важность нулей в шестнадцатеричной системе.

Так как шестнадцатеричная система является позиционной, положение цифры в каждом из разрядов имеет большое значение. Сравним три числа: 0FB, F0B и FB0. Выписав каждое число отдельно и преобразовав каждую цифру шестнадцатеричного числа потетрадно, переведем все три числа в двоичный вид, воспользовавшись снова таблицей 4.1.2.1. Получим:

$$0FB_{16} = 0000\ 1111\ 1011_2 = 251_{10}$$

$$F0B_{16} = 1111\ 0000\ 1011_2 = 3851_{10}$$

$$FB0_{16} = 1111\ 1011\ 0000_2 = 4016_{10}$$

Значит, если нуль в двоичном и шестнадцатеричном числе находится где угодно, кроме крайне левой позиции, то он может крайне сильно поменять значение числа. А вот крайние левые нули (их иногда называют *«начальные»*, или по-английски *leading zeroes*), можно и сократить, – они ни на что не влияют.

4.2. Адреса канального уровня (MAC-адреса)

4.2.1. Одноадресный MAC-адрес

Может показаться несколько излишним внимание к шестнадцатеричной системе и связанными с ней вопросами, но дело в том, что именно шестнадцатеричные цифры используются в записи MAC-адресов.

Для MAC-адреса предусмотрено 48 битов (6 байт), и поэтому читать его в двоичной системе для человека еще труднее, чем IP-адрес: он в полтора раза длиннее. Можно попытаться записать MAC-адрес в десятичной системе счисления, но это 6 октетов, а в такой форме записи его легко перепутать с IP-адресом.

В шестнадцатеричной же системе всего 6 групп по 2 цифры, зачастую еще и с буквами (которые на самом деле тоже цифры).

Рассмотрим такой пример: введем команду ***ipconfig***[1] на компьютере, на котором сейчас пишется этот текст, получим следующий результат, приведенный на рисунке 4.2.1.1.

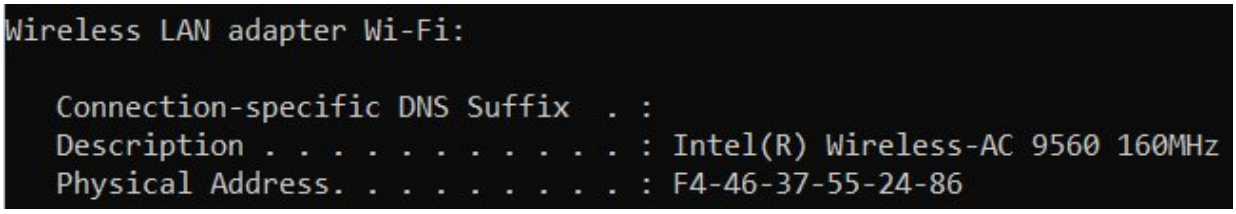

Рисунок 4.2.1.1. – Результат вывода команды ipconfig

На этом рисунке MAC-адрес указан под своим вторым популярным названием – физический адрес (Physical address). Он записан в виде двенадцати цифр в шестнадцатеричном формате, каждая из которых представляет 4 бита. Переведем их в двоичный вид:

Таблица 4.2.1.1. – Перевод MAC-адреса *F4-46-37-55-24-86* в двоичный вид

X_{16}	X_{10}	X_2
F	15	1111
4	4	0100
4	4	0100
6	6	0110
3	3	0011
7	7	0111
5	5	0101
5	5	0101
2	2	0010
4	4	0100
8	8	1000
6	6	0110

Получается, что данное устройство при общении с идентичными ему устройствами будет использовать такую последовательность нулей и единиц:

[1] Это для операционной системы Windows. Для ОС Linux используется команда *ifconfig*

```
11110100010001100011011101010101001001001000011 0
```

Любой MAC-адрес состоит из двух частей:

- OUI – уникальный идентификатор компании-производителя;
- порядковый номер производителя – уникальный номер, который присваивает производитель сетевого адаптера.

Коды OUI выдаются Всемирным советом инженеров электросвязи (IEEE). Все OUI одним списком можно посмотреть по ссылке: https://standards-oui.ieee.org/

```
A8-F9-4B    (hex)              Eltex Enterprise Ltd.
A8F94B      (base 16)          Eltex Enterprise Ltd.
                               Okruzhnaya st. 29v
                               Novosibirsk    630020
                               RU
```

Рисунок 4.2.1.2. – Пример записи о принадлежности OUI

Исходя из того, что на уникальный идентификатор интерфейса отводится 24 бита, можно сделать вывод, что каждый OUI может содержать 2^{24} адресов. Это не мало, но, учитывая количество производимых во всем мире устройств, подключаемых к сети, – крайне недостаточно. Поэтому один производитель может регистрировать на себя несколько OUI по мере исчерпания уже существующих.

У MAC-адреса в первом октете OUI есть два бита, которые несут дополнительную информацию:

1) Предпоследний бит первого октета MAC-адреса указывает на то, является ли данный адрес сконфигурированным самим производителем (0) или подвергся изменениям (1) (устройство может «показывать» сети измененный MAC-адрес в целях безопасности);
2) Последний бит первого октета указывает, предназначен ли данный кадр этого устройства для групповой (1) или одноадресной (0) рассылки.

Рассмотрим пример MAC-адреса нашего устройства:

1111010001000110001101110101010100100100100001 10

Остановимся подробнее на первом октете (байте):

Таблица 4.2.1.2. – Первый октет MAC-адреса *F4-46-37-55-24-86*

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	1	1	1	0	1	**0**	**0**

Заметим, что MAC-адрес рассматриваемого устройства выдан его производителем и не менялся, также на этот адрес будут приниматься только одноадресные рассылки.

Последний момент довольно важен, так как сетевые устройства на пути от источника к получателю по-разному обрабатывают кадры, предназначенные для разного рода рассылок. Поговорим об этом подробнее далее.

4.2.2. Групповой MAC-адрес

Групповой MAC-адрес применяется для рассылки от одного источника данных к множеству получателей. Такой сценарий групповой рассылки можно иллюстрировать

примером трансляций на платформе YouTube. В этом случае блогер, находясь где-то в своей мини-студии, передает контент о чем-то, в чем он, вероятно, хорошо разбирается. В результате тысячи пользователей подключаются к его прямому эфиру. Необходимо отметить, что в данном примере адрес источника будет один, в то время как адресов получателей должно быть много.

В каждом PDU есть только одно поле для адреса источника и для адреса назначения. То есть источник отсылает весь свой групповой трафик на IP-адрес, первый октет у которого лежит в диапазоне 224.0.0.0 – 239.255.255.255, а последующие три октета содержат так называемый номер группы, на который и «подписываются» пользователи (подробно процесс подключения и отключения в этом курсе не рассматривается). В локальной сети этот номер группы должен быть передан группе устройств, чтобы они могли быстро определить, предназначен ли этот трафик для них еще на этапе обработки на *канальном* уровне. Инженеры решили эту проблему: для группового трафика были специально разработаны собственные MAC-адреса.

Групповой MAC-адрес всегда является *адресом назначения (destination address)*, так как групповая рассылка не предусматривает излишней интерактивности: сервер-источник и без этого слишком занят, транслируя большие объемы звука, видео или всего сразу.

Как и одноадресные (*unicast*) MAC-адреса, групповой состоит из двух равных частей, по 24 бита каждая. Однако есть существенные отличия. В первой части вместо *OUI* указывается фиксированное значение – *01-00-5E*. Заметим, что последний бит первого октета равен 1, – это предупреждает устройство получателя о том, что кадр необычный.

Двадцать пятым битом всегда указывается 0. А вот с 26-го по 48-й бит занимают 23 бита из группового IP-адреса. Например, вещание осуществляется из источника групповой рассылки с IP-адресом 239.1.1.15 (адрес гипотетический, все совпадения случайны). Запишем его в двоичном виде (так как в IP-адресе 32 бита, но нужно составить адрес из 48 бит, первые 16 бит не используем).

Таблица 4.2.2.1. – Пример создания группового MAC-адреса. Этап 1

1 байт								2 байт								3 байт								4 байт								5 байт								6 байт							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
																1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1

Теперь запишем первую половину группового MAC-адреса 01-00-5E и 25-й нулевой бит.

Таблица 4.2.2.2. – Пример создания группового MAC-адреса. Этап 2

1 байт								2 байт								3 байт								4 байт								5 байт								6 байт							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0																							

Попробуем совместить:

Таблица 4.2.2.3. – Пример создания группового MAC-адреса. Этап 3

1 байт								2 байт								3 байт								4 байт								5 байт								6 байт							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0																							
																1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1

Видим, что биты с 17-го по 25-й перекрываются. Эту часть IP-адреса мы отбрасываем, и в итоге получается:

В стандартном виде этот MAC-адрес запишется так: 01-00-5E-01-01-0F.

4.2.3. Широковещательный MAC-адрес

Широковещательный MAC-адрес (*broadcast* MAC-*address*) используется для отправки пакетов всем узлам в пределах одной локальной сети (широковещательного домена). Другими словами: все устройства, во-первых, примут этот кадр, во-вторых, его обработают. Если кадр предназначен всем, значит, и каждому устройству в отдельности (рассмотрено в предыдущей главе). Широковещательный MAC-адрес состоит из 48 единиц.

Таблица 4.2.3.1. – Широковещательный MAC-адрес

1 байт								2 байт								3 байт								4 байт								5 байт								6 байт							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Если его записать в шестнадцатеричной системе, получится FF-FF-FF-FF-FF-FF.

Широковещательные рассылки широко используются в среде Ethernet. Два самых известных протокола, использующих широковещательную рассылку – DHCP и ARP (рассмотрены далее в этом курсе).

Широковещательный MAC-адрес также как и групповой MAC-адрес, может быть только адресом назначения. В программе Wireshark, которая позволяет подробно рассматривать трафик, проходящий по какому-либо интерфейсу, широковещательный адрес выглядит так:

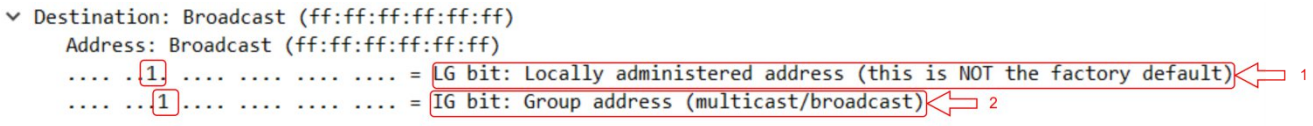

Рисунок 4.2.3.1. – Широковещательный адрес в Wireshark

Обратите внимание на предпоследний и последний биты первого октета: они явно показывают: 1) кадр «придуман» устройством, а не назначен производителем; и 2) относится к групповым или широковещательным адресам.

4.3. Протокол ARP

ARP – протокол, используемый для определения MAC-адреса устройства по его IP-адресу.

4.3.1. ARP-запрос и ARP-ответ

Теперь, когда есть представление о таких понятиях как IP-адрес, MAC-адрес и широковещательная рассылка, познакомимся с протоколом распознавания адресов ARP (англ. *Address Resolution Protocol*), который является одним из важных понятий в сетевом взаимодействии.

ARP – сетевой протокол, используемый для определения MAC-адреса устройства по его IP-адресу

Для передачи данных от одного узла к другому в пределах одной сети узлу-отправителю необходимо знать как IP-адрес, так и MAC-адрес получателя. Без второго адреса протокол Ethernet (канальный уровень) просто не сформирует кадр и не отправит его в среду. Как же отправитель узнает MAC-адрес получателя? Вот как раз для этого и нужен протокол ARP.

Вернемся к модели OSI. Приложение сформировало и отформатировало данные, транспортный уровень разбил данные по сегментам, сетевой уровень присвоил каждому сегменту IP-заголовок, превратив его в пакет. Все готово. Есть даже свой MAC-адрес. Не хватает только MAC-адреса получателя.

Для того чтобы узнать MAC-адрес по IP, узел-отправитель просто спрашивает, какой MAC-адрес у потенциального получателя с определенным IP-адресом.

Например, узел 192.168.1.233 пытается что-то отправить узлу 192.168.1.1:

1) Узел-отправитель посылает широковещательный кадр (FF-FF-FF-FF-FF-FF) с запросом: «У кого IP 192.168.1.1-й? Ответьте 192.168.1.233-му». Этот кадр называется ARP-запрос *(ARP-request)*;

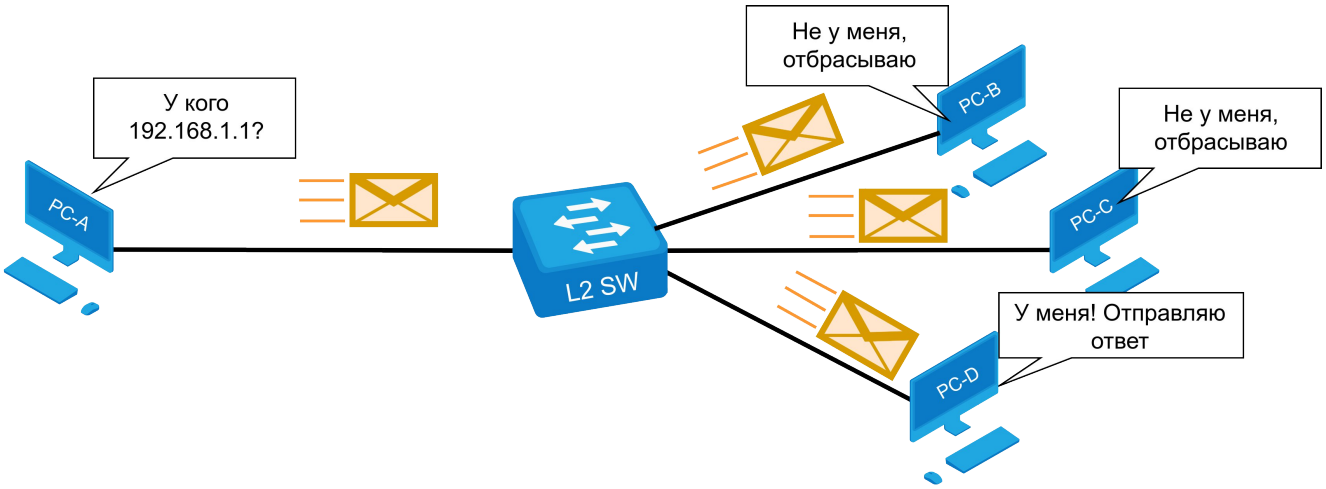

Рисунок 4.3.1.1. – Принцип работы ARP-запроса

2) Все узлы в сети получают этот кадр и распаковывают его.
3) В поле TargetIPAddress протокола ARP узлы видят IP-адрес. Только один узел понимает, что это адресовано ему. Остальные же, отбрасывают кадр, так как он предназначен не им.
4) Тот единственный узел с адресом 192.168.1.1 понимает, что он должен ответить.
5) Он формирует кадр, и отправляет его обратно на PC-A.
6) PC-A записывает полученный MAC адрес в свою ARP таблицу.

Благодаря такому протоколу, устройства, находящиеся в одной сети, могут узнать необходимый им MAC адрес.

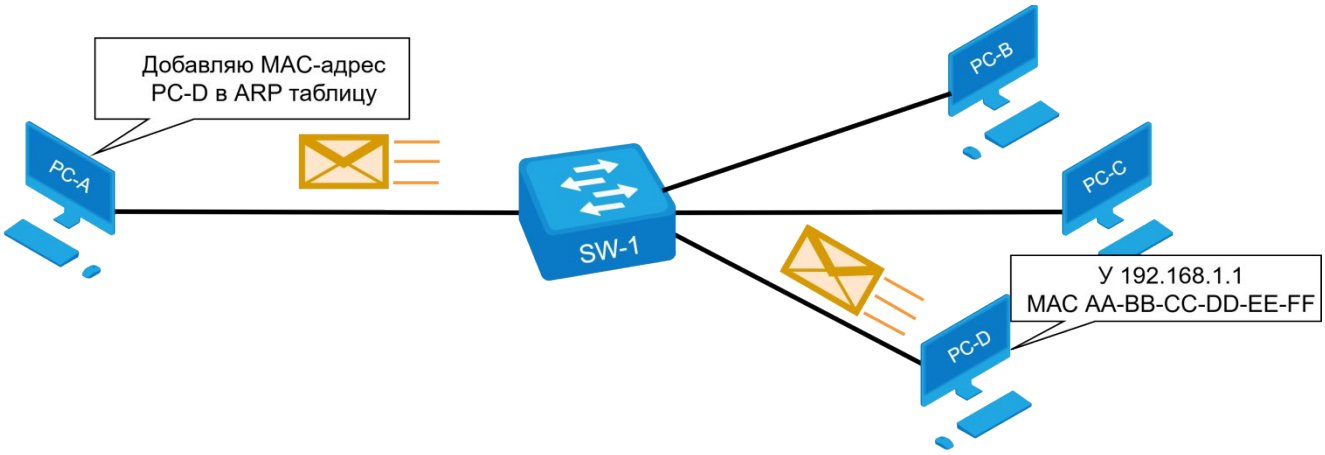

Рисунок 4.3.1.2. – Принцип работы ARP-ответа

Теперь подробнее узнаем как происходит этот процесс с помощью ПО Wireshark.

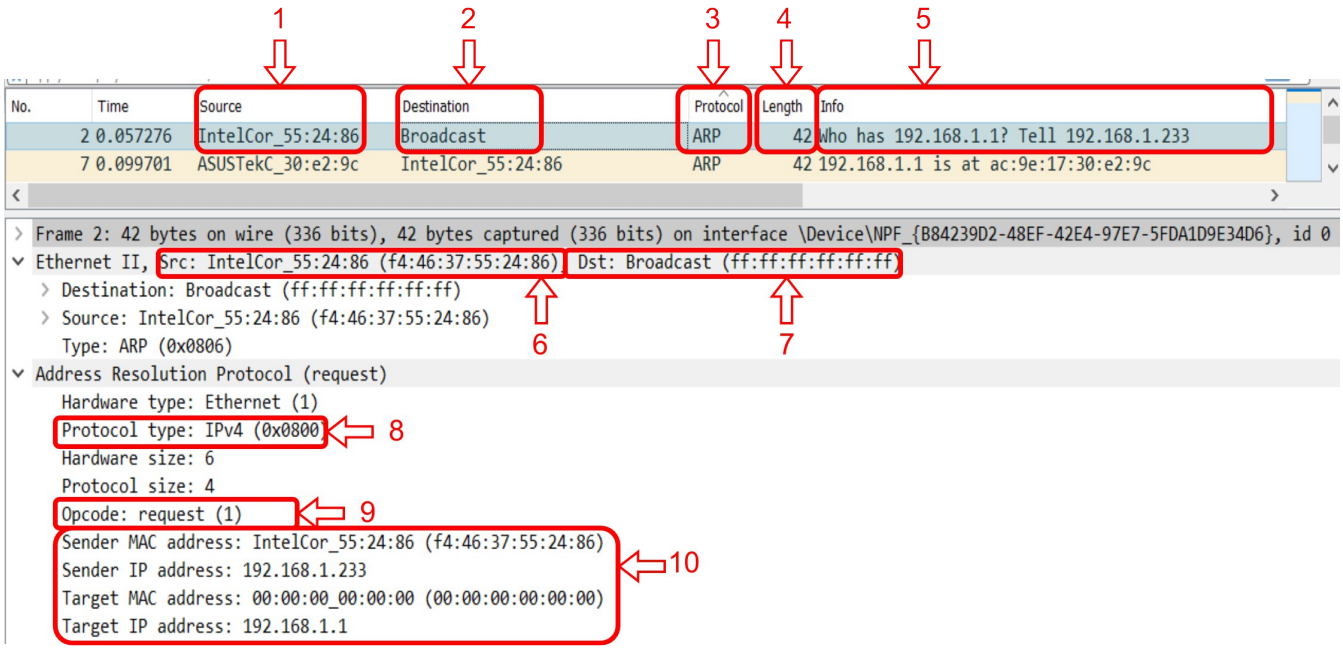

Рисунок 4.3.1.3. – Представление ARP-запроса в Wireshark

Рассмотрим некоторые поля, которые помогут нам разобраться в происходящем:

1. Source (источник). В данном поле программа показывает нам с какого MAC-адреса был сделан запрос. То есть, физический адрес компьютера, которому необходимо узнать MAC-адрес другого компьютера в сети. Адрес представлен в не совсем привычной форме, и первая половина (24 бита) уже преобразованы в имена компаний-производителей.
2. Destination (назначение). В поле адреса назначения мы видим слово Broadcast, что означает широковещательную рассылку на все MAC-адреса в рамках этой сети. Широковещательная рассылка используется потому, что компьютер отправителя не знает физический адрес назначения, и «спрашивает» его у всех доступных устройств.
3. Protocol (протокол). Здесь отображается используемый протокол. ARP в нашем случае.

4. Length (длина). Указывается общий размер пакета в байтах. Текущий составляет 42 байта.
5. Info (информация). Это поле отображает краткую информацию о содержимом каждого пакета. Здесь мы видим дословный запрос: У кого адрес 192.168.1.1? Ответьте на адрес 192.168.1.233.
6. Src (source). Здесь мы видим то же самое поле с MAC-адресом источника, только уже в полном виде.
7. Dst (Destination). Аналогичное поле с MAC-адресом назначения.
8. Protocol type (тип протокола). Указывается тип используемого протокола. В нашем случае это IPv4, возможен вариант с IPv6.
9. Opcode (код операции). Данное поле в рамках протокола ARP показывает нам, что, если стоит 1 – это запрос, если 2 – это ответ.
10. Sender (Отправитель) и Target (цель, получатель). Данные поля показывают нам MAC и IP-адреса отправителя и получателя. MAC-адрес получателя указан как 00:00:00:00:00:00, потому что он еще неизвестен.

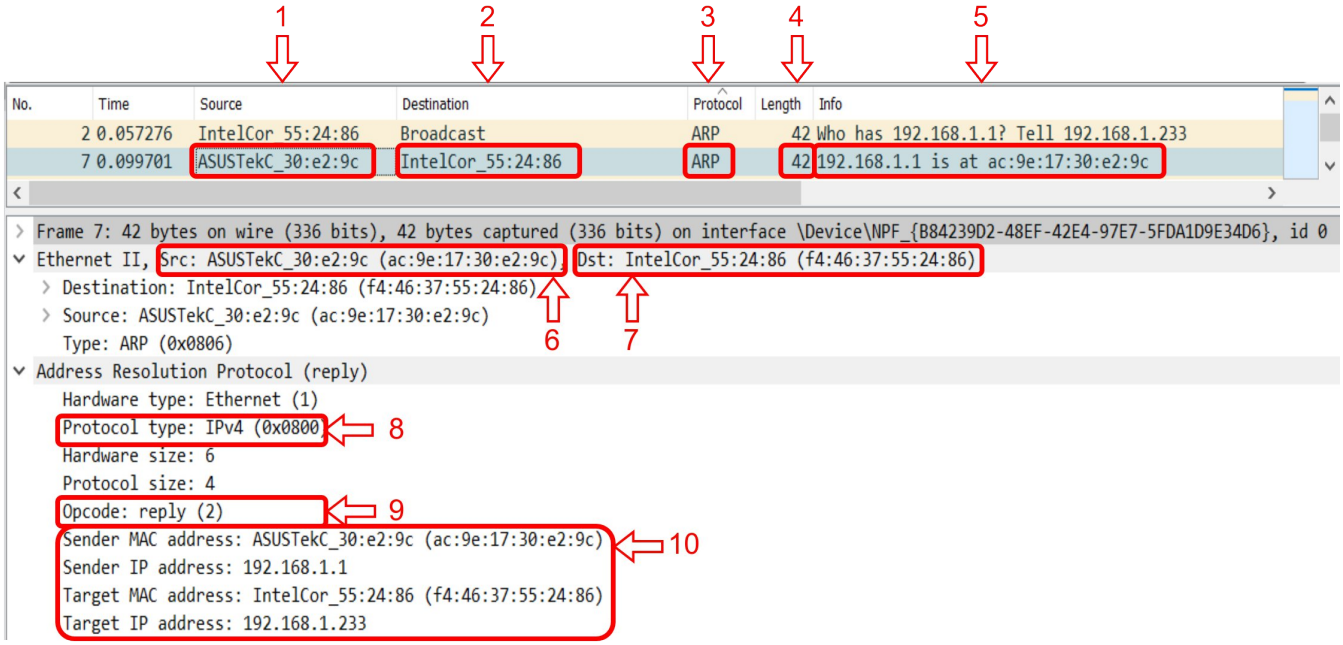

Рисунок 4.3.1.4. – ARP-ответ в Wireshark

Когда приходит ARP ответ, ситуация несколько изменяется.

1. Source (источник). Здесь физическим адресом источника становится искомый адрес компьютера.
2. Destination (назначение). Адресом назначения у нас становится предыдущий MAC-адрес источника, так как ответить нужно на него.
3. Protocol (протокол). Без изменений.
4. Length (длина). Указывается общий размер пакета в байтах. Текущий составляет 42 байта.
5. Info (информация). В информации мы можем наблюдать ответ. 192.168.1.1 имеет

физический адрес ac:9e:17:30:e2:9c

6. Src (source). Адрес источника также изменился, так как он стал известен.
7. Dst (Destination). Адрес назначения стал предыдущим адресом источника, так как это ответ.
8. Protocol type (тип протокола). Без изменений.
9. Opcode (код операции). Как видно, цифра сменилась на 2, значит, это ARP-ответ.
10. Sender (Отправитель) и Target (цель, получатель). В данных полях мы уже не видим адреса 00:00:00:00:00:00, так как отправителю известен свой адрес, и за счёт предыдущего сообщения известен адрес получателя.

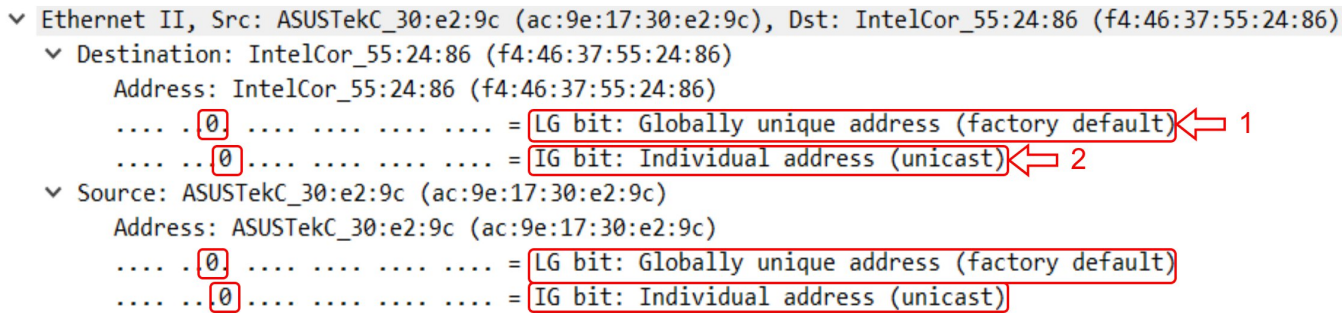

Рисунок 4.3.1.5. – Одноадресные MAC-адреса в ARP-ответе

На рисунке 4.3.1.5 Представлены развернутые поля Destination и Source. Здесь важно отметить, что данный MAC-адрес является оригинальным (то есть не подменным). На это нам указывает бит 0 в строке под номером 1. В строке 2 мы можем увидеть, что рассылка выполняется в одноадресном формате, так как мы точно знаем адрес назначения.

4.3.2. Таблица ARP-соответствий

Процесс постоянной рассылки широковещательных сообщений – дело накладное в плане трафика и излишней загрузки ни к чему не причастных узлов обработкой трафика. Поэтому, чтобы не рассылать ARP-запросы постоянно, узел вносит результаты ARP-ответов в таблицу известных сочетаний IP- и MAC-адресов, так называемую ARP-таблицу.

На рисунке 4.3.2.1 представлена ARP-таблица узла под управлением операционной системы Windows.

Теперь, если узлу необходимо отправить какой-то пакет на известный ему IP-адрес, он в первую очередь проверит наличие соответствующего ему MAC-адреса в ARP-таблице и только потом отправит широковещательный ARP-запрос.

```
Interface: 192.168.1.233 --- 0x16
  Internet Address      Physical Address      Type
  192.168.1.1           ac-9e-17-30-e2-9c     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Рисунок 4.3.2.1. – Таблица ARP-соответствий в операционной системе Windows

Для хранения соответствий в ARP-таблице узел использует определенные ресурсы, поэтому такая таблица не может быть бесконечной: через определенное время соответствия стираются, и тогда узел должен будет снова прибегнуть к рассылке ARP-запроса для выявления нужного в целях пересылки MAC-адреса.

4.4. Обмен данными внутри одной сети

Будучи вооруженными знаниями из этой и предыдущих глав учебника, представим, как происходит обмен данными внутри одной локальной сети. Рассмотрим, как компьютер PC-A отправляет некий файл компьютеру PC-B[2].

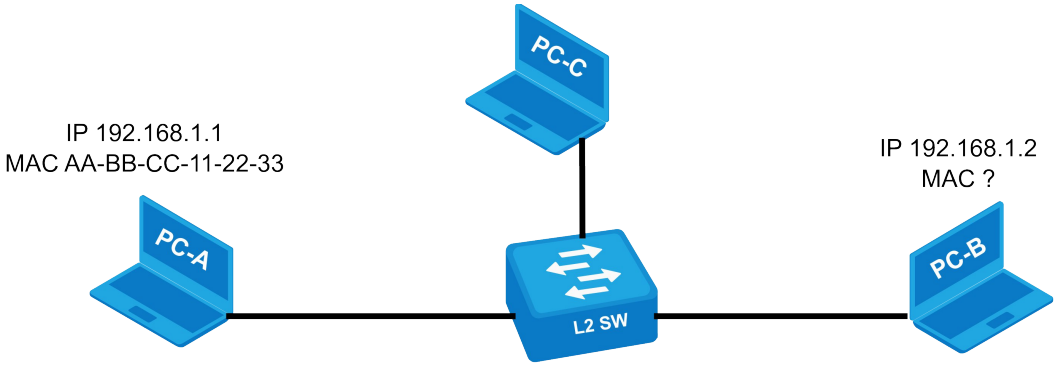

Рисунок 4.4.1. – Пример локальной сети для передачи данных

На первом этапе на ***уровне приложений*** информация о передаваемом файле поступает к протоколу FTP. Он производит форматирование файла для отправки и передает эти данные на уровень ниже – протоколу TCP.

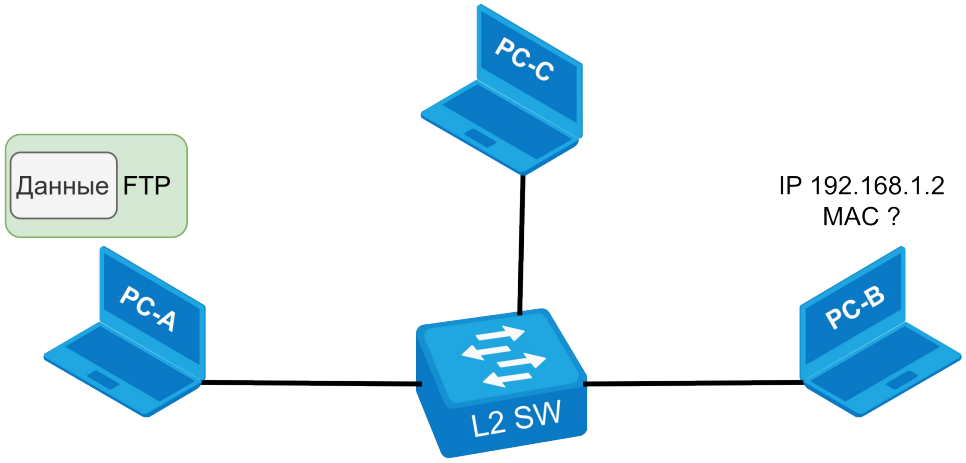

Рисунок 4.4.2. – Пример локальной сети для передачи данных. Процесс инкапсуляции данных для передачи на

[2] Для упрощения понимания, до тех пор, пока мы не познакомились подробно с работой протокола TCP, будем считать, что весь файл «поместится» в один сегмент.

Протокол TCP разделяет файл на сегменты и каждому присваивает метку – номер порта источника (какое приложение передает данные) и номер назначения (какому приложению оно предназначается). В данном случае порту источника будет присвоен 54236, а порт приемника будет иметь номер 21 (закреплен за протоколом FTP).

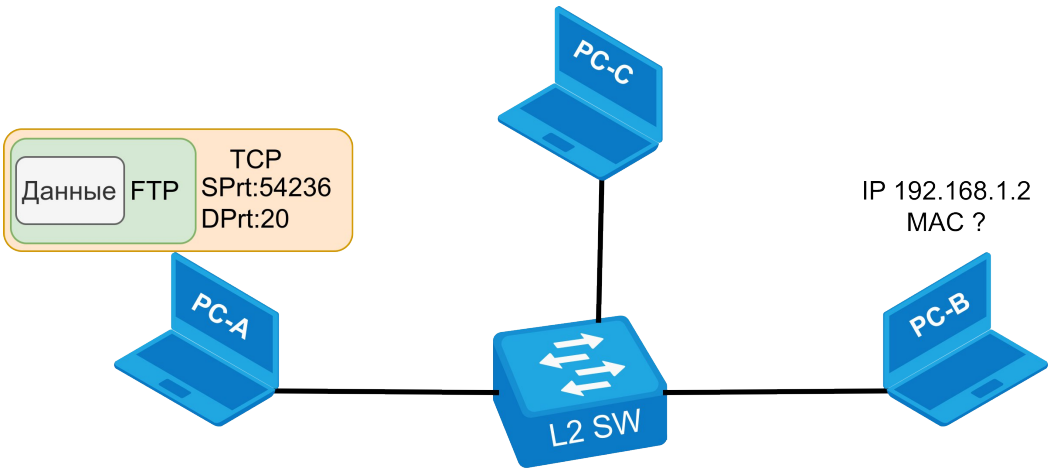

Рисунок 4.4.3. – Пример локальной сети для передачи данных. Процесс инкапсуляции данных в сегмент для передачи на сетевой уровень

Далее сегмент передается на сетевой уровень к протоколу IP, который «проставляет» IP-адреса отправителя (SrcIP) и получателя (DstIP). Он может это сделать, так как компьютер-отправитель знает и свой IP-адрес, и IP-адрес получателя.

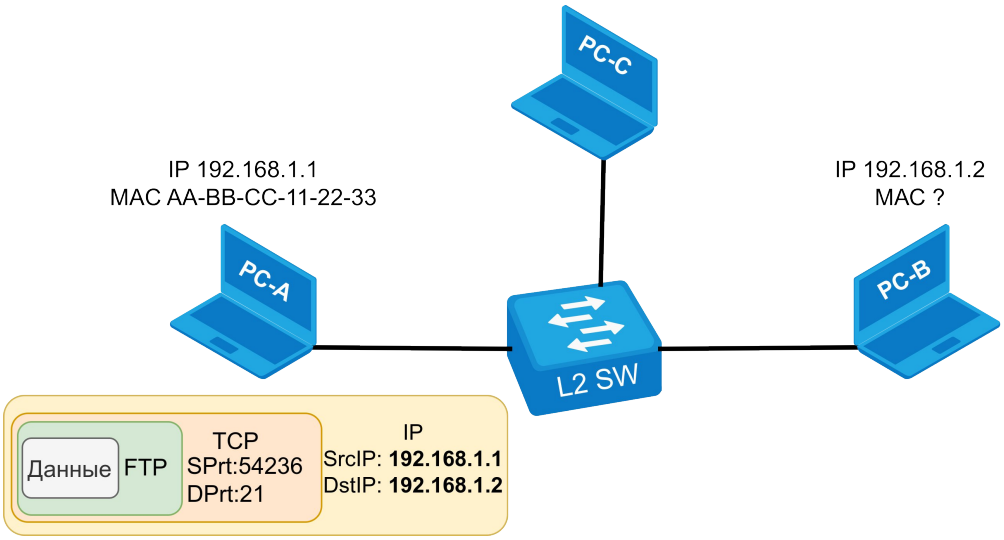

Рисунок 4.4.4. – Пример локальной сети для передачи данных. Формирование IP-пакета на сетевом уровне

На этом этапе PC-A вынужден прерваться, так как он не знает MAC-адреса PC-B. Поэтому он запоминает пакет и формирует ARP-запрос, в котором указывает в качестве IP-адреса получателя IP-адрес PC-B, а в качестве MAC-адреса получателя – широковещательный адрес FF:FF:FF:FF:FF:FF.

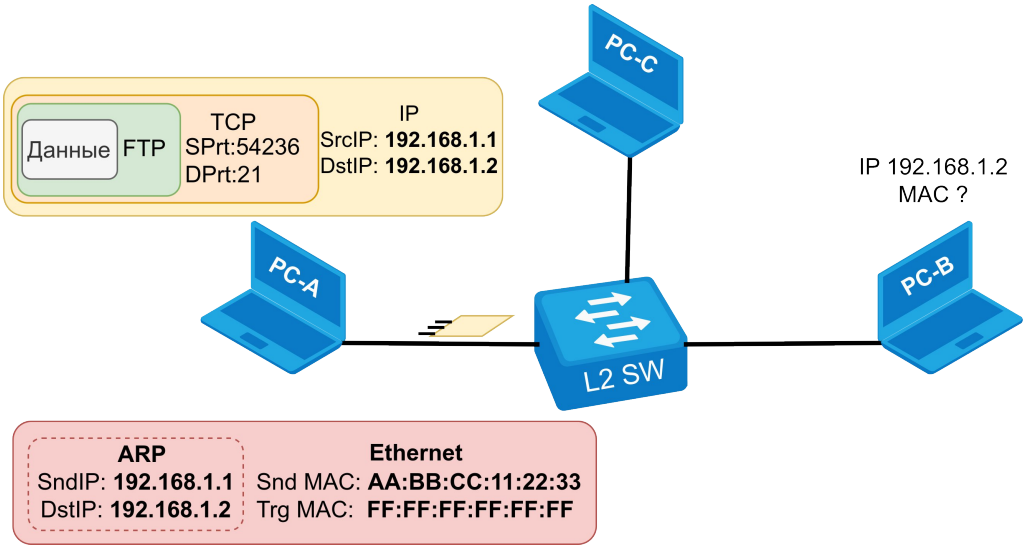

Рисунок 4.4.5. – Пример локальной сети для передачи данных. Формирование ARP-запроса

Так как PC-A отправляет кадр на широковещательный MAC-адрес, коммутатор L2 SW отправляет его всем подключенным к нему устройствам.

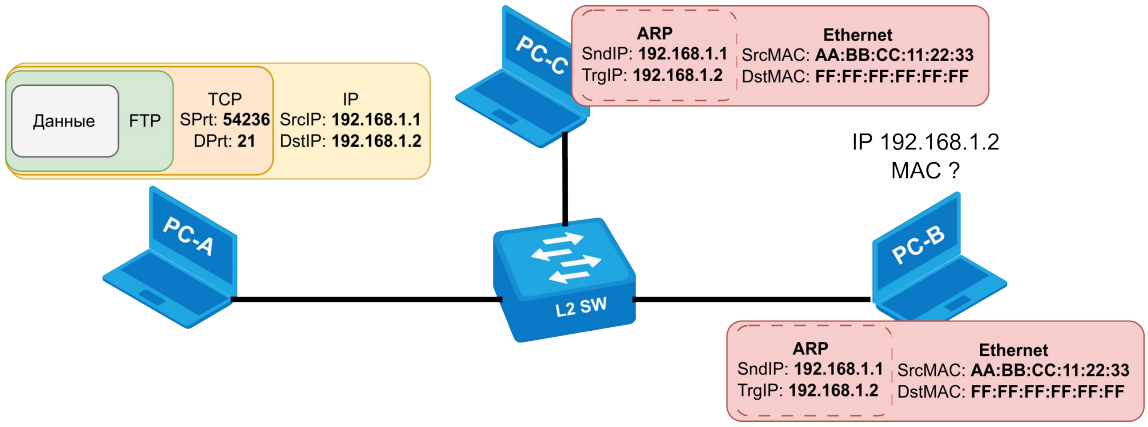

Рисунок 4.4.6. – Пример локальной сети для передачи данных. Широковещательная рассылка ARP-запроса

Получив широковещательный кадр, PC-B и PC-C отбрасывают заголовок Ethernet и сравнивают IP-адрес получателя со своим собственным. При этом PC-C убеждается, что данное сообщение предназначается не ему, и отбрасывает пакет ARP, а PC-B принимает его и использует его информацию для ARP-ответа.

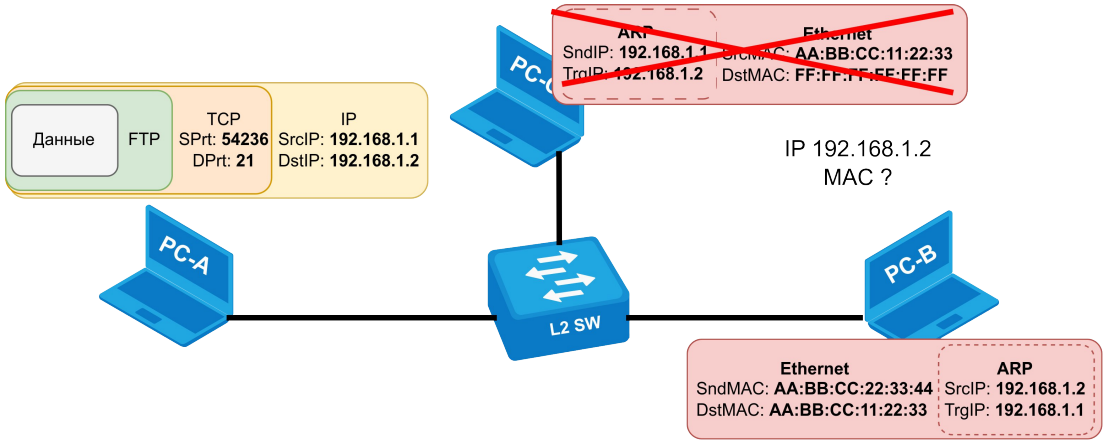

Рисунок 4.4.7. – Пример локальной сети для передачи данных. Одноадресная отправка ARP-ответа от PC-B к PC-A

PC-A получает ARP-ответ от PC-B, записывает MAC-адрес из поля SrcMAC в таблицу ARP, а также использует его для доформирования пакета, который он все это время хранил в памяти[3].

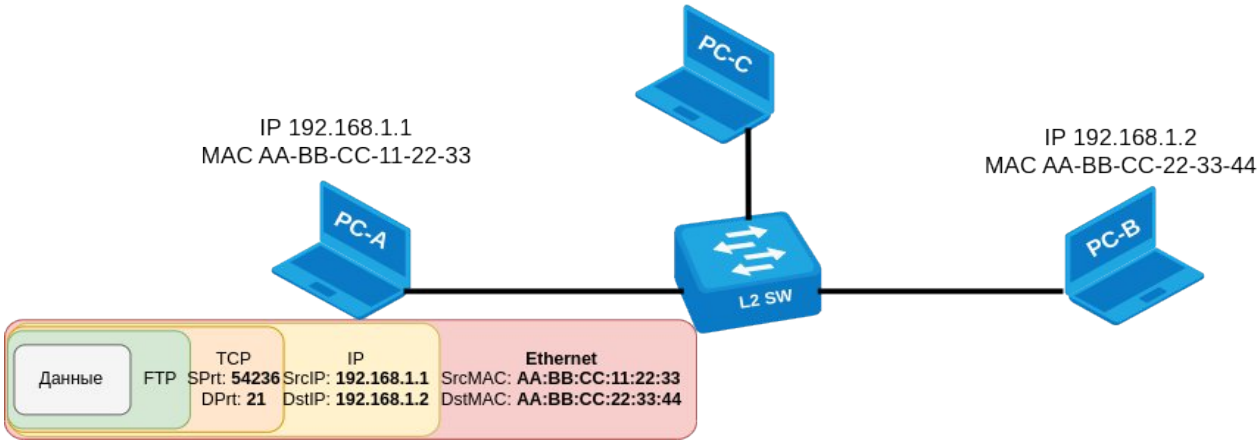

Рисунок 4.4.8. – Пример локальной сети для передачи данных. Процесс инкапсуляции пакета в кадр для передачи на канальный уровень

Компьютер PC-A отправляет сформированный кадр посредством одноадресной рассылки компьютеру PC-B.

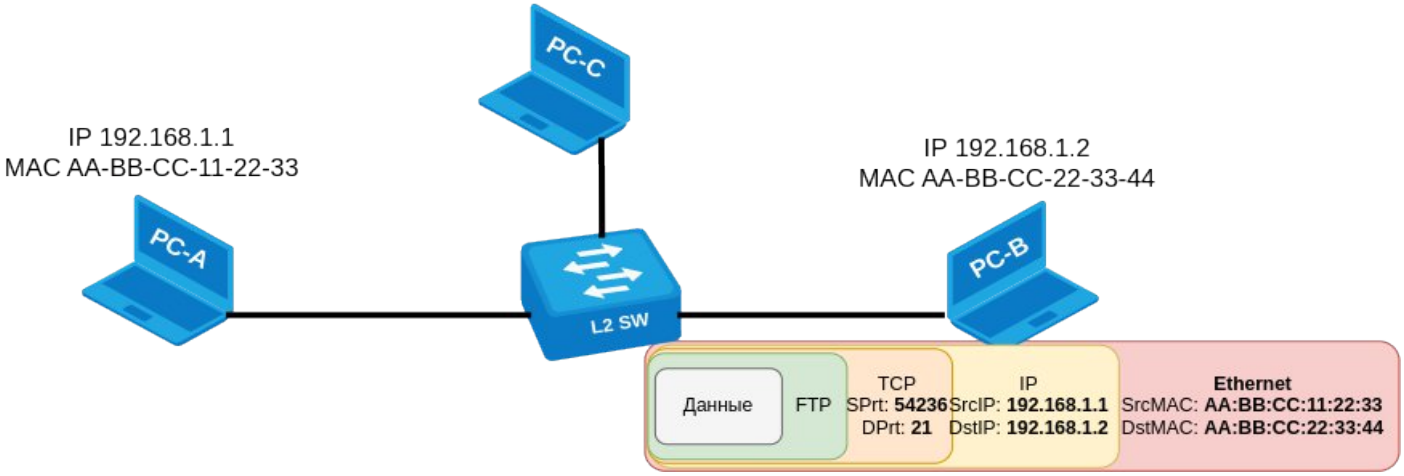

Рисунок 4.4.9. – Пример локальной сети для передачи данных. Процесс приема кадра PC-B от PC-A

Компьютер PC-B получает кадр, убеждается, что MAC-адрес назначения совпадает с его собственным, удаляет заголовок Ethernet и передает полученный пакет на уровень выше.

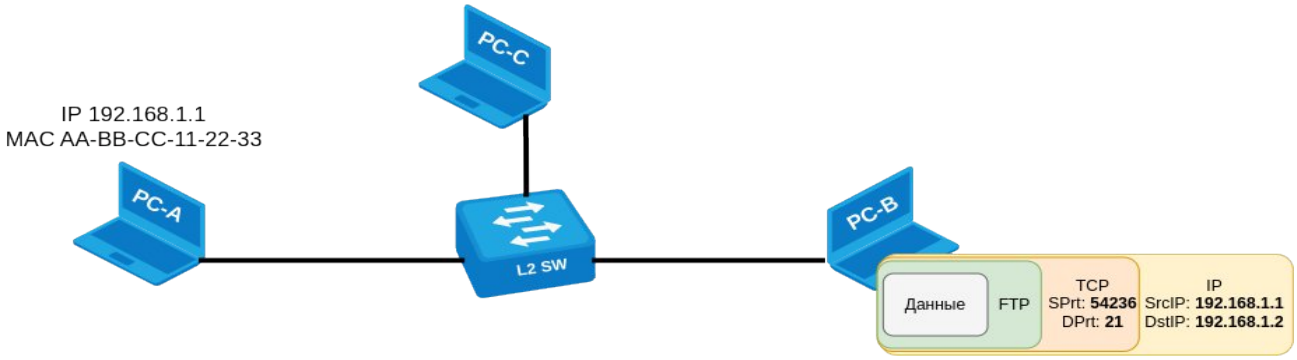

Рисунок 4.4.10. – Пример локальной сети для передачи данных. Принятый кадр декапсулирован в пакет

[3] На самом деле, не всегда. Иногда у таких пакетов истекает «срок годности» и приложению приходится делать еще одну попытку отправить данные, но к тому времени у него уже будет нужный MAC-адрес из ARP-таблицы.

Далее он сравнивает DstIP со своим собственным, удаляет заголовок IP передает полученный сегмент на уровень выше.

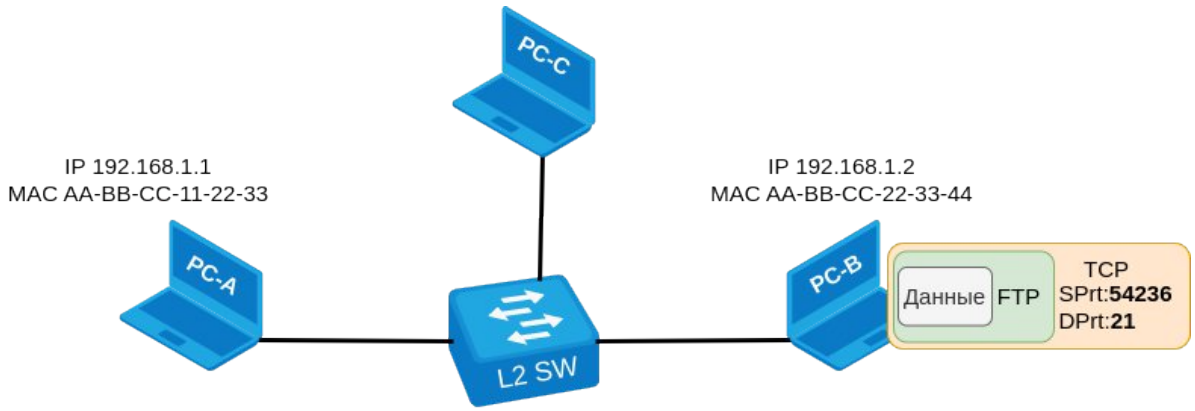

Рисунок 4.4.11. – Пример локальной сети для передачи данных. Принятый пакет декапсулирован в сегмент

Данные из поля «порт назначения» (DPrt) используются для идентификации приложения или протокола. Порт 21 принадлежит протоколу FTP. Удаляется заголовок сегмента, и данные передаются протоколу FTP.

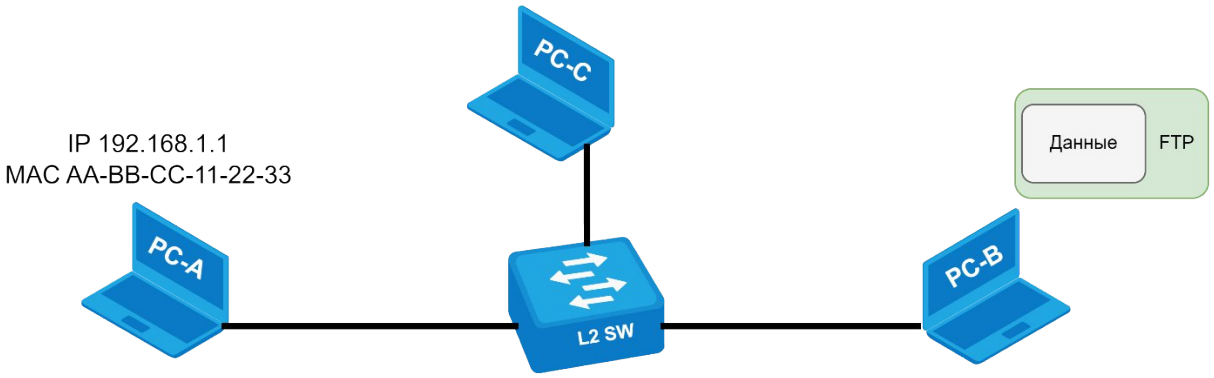

Рисунок 4.4.12. – Пример локальной сети для передачи данных. Удаление заголовка сегмента, отправка данных на уровень приложений

Протокол FTP дешифрует и форматирует данные, после чего записывает их на жесткий диск.