

Университет ИТМО
Факультет программной инженерии и компьютерной техники

Лабораторная работа №2
«Локальные сети»
по дисциплине «Компьютерные сети»

Выполнил:
Студент 3 курса группы Р3331
Дворкин Борис Александрович

Преподаватель:
Алиев Тауфик Измайлович

г. Санкт-Петербург

2024 г.

Содержание

Содержание.....	2
1 Введение.....	3
2 Вариант лабораторной работы.....	3
3 Этап 1. Локальная сеть с концентратором.....	3
3.1 Построение сети.....	3
3.2 Анализ таблиц.....	4
ARP-таблица.....	4
Таблица маршрутизации.....	5
3.3 Тестирование сети (отправка пакетов по UDP).....	6
3.3.1. Какие пакеты и кадры передаются в сети?.....	7
3.3.2. Как происходит передача, что содержится в пакетах?.....	8
3.3.3. Появились ли изменения в таблицах?.....	8
3.4 Тестирование сети (отправка пакетов по TCP).....	9
1. ARP-обмен (определение MAC-адресов).....	10
2. Установление TCP-соединения (трехэтапное рукопожатие).....	10
3. Передача данных.....	11
4. Завершение соединения (четырёхэтапное рукопожатие).....	12
4 Этап 2. Локальная сеть с коммутатором.....	13
4.1 Построение сети.....	13
4.2 Анализ таблиц.....	13
4.3 Тестирование сети (отправка пакетов).....	14
5 Этап 3. Многосегментная локальная сеть.....	15
5.1 Построение сети.....	15
5.2 Тестирование сети (отправка пакетов), анализ таблиц.....	15
6 Выводы.....	16

1 Введение

Целью работы является изучение принципов настройки и функционирования локальных сетей, построенных с использованием концентраторов и коммутаторов, а также процессов передачи данных на основе стека протоколов TCP/IP, с использованием программы моделирования компьютерных сетей NetEmul.

2 Вариант лабораторной работы

Ф = 7 (Дворкин); И = 5 (Борис); О = 13 (Александрович); Н = 31 (Р3331)

=> класс А: $(7 + 31).(5 + 31).(13 + 31).(7 + 5) \Leftrightarrow 38.36.44.12$

=> класс В: $(5 + 31 + 128).(13 + 31).(7 + 31).(7 + 5) \Leftrightarrow 164.44.38.12$

=> класс С: $(192 + 31 + 13).(7 + 31).(5 + 31).(7+5) \Leftrightarrow 236.38.36.12$

Порядковый номер в группе: 4

=> N1 = 2; N2 = 3; N3 = 3; Класс IPv4-адресов: В

(А я захотел N1 = 4; N2 = 6; N3 = 4; Вопросы?)

3 Этап 1. Локальная сеть с концентратором

3.1 Построение сети

Для нумерации интерфейсов двух компьютеров в данной сети используется пул последовательных адресов: **164.44.38.12 - 164.44.38.15**

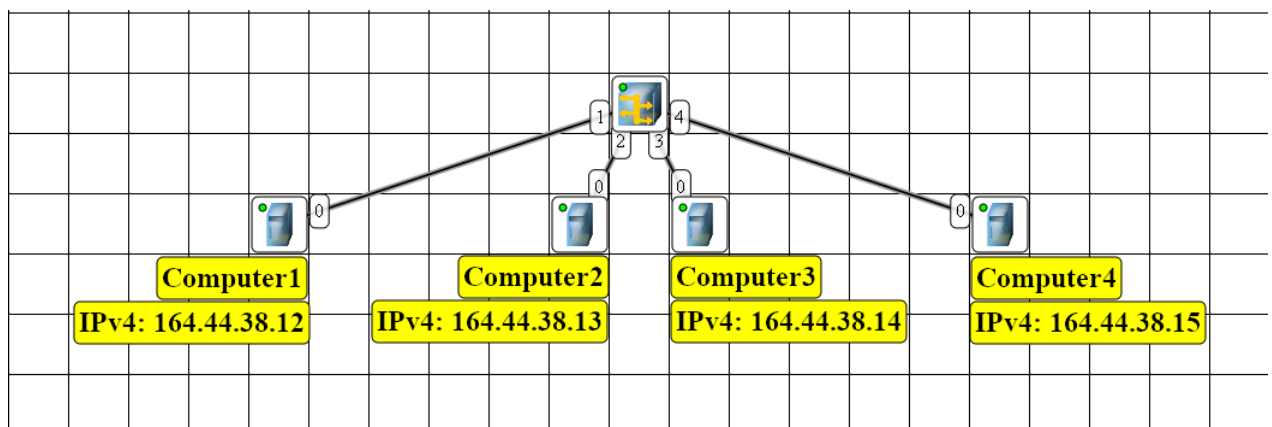


Рис.1: Модель локальной сети с концентратором

3.2 Анализ таблиц

ARP-таблица

Address Resolution Protocol - протокол для разрешения MAC-адресов физического сетевого устройства по IP-адресу компьютера. ARP-таблица - хранит соответствия между IP-адресами и MAC-адресами устройств в локальной сети. Это позволяет устройствам находить друг друга на канальном уровне (уровне сетевого интерфейса в TCP/IP). Если хотим наоборот получить IP-адрес по MAC-адресу, то можем воспользоваться RARP (Reverse ARP) протоколом.

До отправки ARP-запросов, с устройств Computer 1-4 ARP таблицы пустые.

После отправки данных по протоколу UDP произошёл ряд ARP-запросов и таблицы Computer1-4 заполнились записями о соответствии MAC-адресов сетевых интерфейсов и IPv4-адресов 3х других компьютеров:

- Computer1

	Mac-address	Ip-address	Record type	Netcard name	TTL
1	01:E3:00:4E:A7:45	164.44.38.15	Dinamic	eth0	311
2	01:4D:C9:0A:97:B1	164.44.38.13	Dinamic	eth0	223
3	01:11:13:17:9F:3E	164.44.38.14	Dinamic	eth0	218

- Computer2

	Mac-address	Ip-address	Record type	Netcard name	TTL
1	01:68:AE:60:AD:C8	164.44.38.12	Dinamic	eth0	490
2	01:E3:00:4E:A7:45	164.44.38.15	Dinamic	eth0	476
3	01:11:13:17:9F:3E	164.44.38.14	Dinamic	eth0	288

- Computer3

	Mac-address	Ip-address	Record type	Netcard name	TTL
1	01:68:AE:60:AD:C8	164.44.38.12	Dinamic	eth0	388
2	01:E3:00:4E:A7:45	164.44.38.15	Dinamic	eth0	503
3	01:4D:C9:0A:97:B1	164.44.38.13	Dinamic	eth0	321

- Computer4

	Mac-address	Ip-address	Record type	Netcard name	TTL
1	01:68:AE:60:AD:C8	164.44.38.12	Dinamic	eth0	534
2	01:4D:C9:0A:97:B1	164.44.38.13	Dinamic	eth0	338
3	01:11:13:17:9F:3E	164.44.38.14	Dinamic	eth0	333

Пояснения по содержимому таблиц:

- ARP-запрос был отправлен с Computer 1 и получен всеми устройствами в сети, в данном случае - Computer2-4. Соответствующие компьютеры - сначала Computer2, а затем и 3-4 при последующих запросах узнали в запросе свой IP-адрес и отправили ARP-ответ, после чего MAC-адрес Computer 2-4 был **закеширован** в ARP-таблице Computer 1, чтобы не запрашивать его каждый раз (аналогично с другими компьютерами).
- Записи являются *динамическими*, то есть они создаются автоматически при отправке ARP-запросов и удаляются через определенное *время жизни* (TTL - Time To Live).

Таблица маршрутизации

- содержит информацию о том, как данные должны передаваться между сетями. Она определяет, куда отправлять пакеты в зависимости от их IP-адреса назначения.

Таблица маршрутизации Computer 1:

	Destination	Mask	Gateway	Interface	Metric	Source
1	164.44.0.0	255.255.0.0	164.44.38.12	164.44.38.12	0	Connected

Таблица маршрутизации Computer 2:

	Destination	Mask	Gateway	Interface	Metric	Source
1	164.44.0.0	255.255.0.0	164.44.38.13	164.44.38.13	0	Connected
2	127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	0	Connected

Таблица маршрутизации Computer 3:

	Destination	Mask	Gateway	Interface	Metric	Source
1	164.44.0.0	255.255.0.0	164.44.38.14	164.44.38.14	0	Connected
2	127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	0	Connected

Таблица маршрутизации Computer 4:

	Destination	Mask	Gateway	Interface	Metric	Source
1	164.44.0.0	255.255.0.0	164.44.38.15	164.44.38.15	0	Connected

Пояснения по содержимому таблиц (на примере Computer1):

1. Назначение (Destination):

- Указывает сеть или IP-адрес, куда должен быть отправлен пакет. В нашем случае это локальная сеть **164.44.0.0**.

2. Маска (Mask):

- Определяет, какая часть IP-адреса относится к сети, а какая — к узлу. Маска **255.255.0.0** означает, что первые 16 битов (2 октета) относятся к сети **mina B**, а последние 16 бит — к узлу (транжирство).

3. Шлюз (Gateway):

- Это IP-адрес устройства, через которое пакеты должны быть отправлены, если они не находятся в локальной сети. В данном случае шлюз указывает на интерфейс самого устройства.

4. Интерфейс (Interface):

- Это сетевой интерфейс, через который пакеты будут отправлены. В нашем случае это Ethernet-интерфейс **eth0** с IP-адресом **164.44.38.12**.

5. Метрика (Metric):

- Это числовое значение, которое определяет приоритет маршрута. Чем меньше метрика, тем предпочтительнее маршрут. В нашем случае метрика равна 0, что означает, что это прямой маршрут (устройства находятся в одной сети).

6. Источник (Source):

- Состояние источника. Connected - подключено.

3.3 Тестирование сети (отправка пакетов по UDP)

computer1	Computer2	Computer3	Computer4
<pre>> sent 164.44.38.12 search 164.44.38.13 Type: ARP request > received 164.44.38.12 found 164.44.38.13 Type: ARP response > sent 164.44.38.12 >> 164.44.38.13 Type: UDP Message user Ethernet, sender: 01:68:AE:60:AD:C8 receiver: 01:4D:C9:0A:97:B1 IP packet, sender: 164.44.38.12, receiver: 164.44.38.13 TTL: 64 UDP, sender port: 7777, receiver port: 7777</pre>	<pre>> received 164.44.38.12 search 164.44.38.13 Type: ARP request > sent 164.44.38.12 found 164.44.38.13 Type: ARP response > received 164.44.38.12 >> 164.44.38.13 Type: UDP Message user Ethernet, sender: 01:68:AE:60:AD:C8 receiver: 01:4D:C9:0A:97:B1 IP packet, sender: 164.44.38.12, receiver: 164.44.38.13 TTL: 64 UDP, sender port: 7777, receiver port: 7777</pre>	<pre>> received 164.44.38.12 search 164.44.38.13 Type: ARP request > sent 164.44.38.12 found 164.44.38.13 Type: ARP response > received 164.44.38.12 >> 164.44.38.13 Type: UDP Message user Ethernet, sender: 01:68:AE:60:AD:C8 receiver: 01:4D:C9:0A:97:B1 IP packet, sender: 164.44.38.12, receiver: 164.44.38.13 TTL: 64 UDP, sender port: 7777, receiver port: 7777</pre>	<pre>> received 164.44.38.12 search 164.44.38.13 Type: ARP request > sent 164.44.38.12 found 164.44.38.13 Type: ARP response > received 164.44.38.12 >> 164.44.38.13 Type: UDP Message user Ethernet, sender: 01:68:AE:60:AD:C8 receiver: 01:4D:C9:0A:97:B1 IP packet, sender: 164.44.38.12, receiver: 164.44.38.13 TTL: 64 UDP, sender port: 7777, receiver port: 7777</pre>

Рис.2: Журнал сообщений Computer 1 и Computer 2

3.3.1. Какие пакеты и кадры передаются в сети?

1. ARP-запрос (ARP request):

- Это широковещательный запрос, который отправляет Computer1, чтобы узнать MAC-адрес устройства с IP-адресом 164.44.38.13 (Computer2).
- Кадр Ethernet:
 - Отправитель (Sender): 01:68:AE:60:AD:C8 (MAC-адрес Computer1).
 - Получатель (Receiver): FF:FF:FF:FF:FF:FF (широковещательный адрес, запрос отправляется всем устройствам в сети).
- ARP-запрос:
 - IP-адрес отправителя: 164.44.38.12 (Computer1).
 - MAC-адрес отправителя: 01:68:AE:60:AD:C8 (Computer1).
 - IP-адрес назначения: 164.44.38.13 (Computer2).
 - MAC-адрес назначения: 00:00:00:00:00:00 (неизвестен, поэтому запрашивается).

2. ARP-ответ (ARP response):

- Это ответ от Computer2 на ARP-запрос, в котором он сообщает свой MAC-адрес.
- Кадр Ethernet:
 - Отправитель (Sender): 01:4D:C9:0A:97:B1 (MAC-адрес Computer2).
 - Получатель (Receiver): 01:68:AE:60:AD:C8 (MAC-адрес Computer1).
- ARP-ответ:
 - IP-адрес отправителя: 164.44.38.13 (Computer2).
 - MAC-адрес отправителя: 01:4D:C9:0A:97:B1 (Computer2).
 - IP-адрес назначения: 164.44.38.12 (Computer1).
 - MAC-адрес назначения: 01:68:AE:60:AD:C8 (Computer1).

3. UDP-пакет (UDP Message user):

- Это пакет данных, который Computer 1 отправляет на Computer 2 с использованием протокола UDP.
- Кадр Ethernet:
 - Отправитель (Sender): 01:68:AE:60:AD:C8 (MAC-адрес Computer1).

- Получатель (Receiver): 01:4D:C9:0A:97:B1 (MAC-адрес Computer2).
- IP-пакет:
 - Отправитель (Sender): 164.44.38.12 (Computer1).
 - Получатель (Receiver): 164.44.38.13 (Computer2).
 - TTL (Time To Live): 64 (время жизни пакета, уменьшается на каждом маршрутизаторе).
- UDP-сегмент:
 - Порт отправителя (Sender port): 7777.
 - Порт получателя (Receiver port): 7777.

3.3.2. Как происходит передача, что содержится в пакетах?

Сначала ARP-запрос, чтобы узнать MAC-адрес Computer 2, затем ARP-ответ от Computer 2 с указанием своего MAC-адреса, и после получения ARP-ответа устройством Computer 1, отправка UDP-пакета от Computer 1 до Computer 2 с Ethernet пакетом, содержащим MAC-адреса отправителя и получателя, UDP сегментом с портами отправителя и получателя и IP пакет с IP-адресами отправителя и получателя.

При этом примечательно, что при посылке данных из Computer1 до Computer2 - пакеты приходят и в Computer3 и в Computer4!

Это обусловлено принципом работы концентратора - он берёт пакет пришедший на один сетевой интерфейс и перенаправляет его на все остальные сетевые интерфейсы.

3.3.3. Появились ли изменения в таблицах?

В таблицах маршрутизации, ожидаемо, изменений не произошло, т.к. конфигурация (топология) сети не поменялась, однако содержимое ARP-таблиц обновилось записями о соответствии MAC-адресов и IP-адресов Computer 1-4.

3.4 Тестирование сети (отправка пакетов по TCP)

Computer1	Computer2
<ul style="list-style-type: none">sent 164.44.38.12 search 164.44.38.13 Type: ARP request<ul style="list-style-type: none">Ethernet, sender: 01:68:AE:60:AD:C8 receiver: FF:FF:FF:FF:FF:FF> ARP-request:received 164.44.38.12 found 164.44.38.13 Type: ARP response<ul style="list-style-type: none">Ethernet, sender: 01:4D:C9:0A:97:B1 receiver: 01:68:AE:60:AD:C8> ARP-response:sent 164.44.38.12 >> 164.44.38.13 Type: TCP<ul style="list-style-type: none">Ethernet, sender: 01:68:AE:60:AD:C8 receiver: 01:4D:C9:0A:97:B1IP packet, sender: 164.44.38.12, receiver: 164.44.38.13 TTL: 64TCP, sender port: 7777, receiver port: 7777<ul style="list-style-type: none">ISN 3663, ACK 0flags: SYNreceived 164.44.38.13 >> 164.44.38.12 Type: TCP<ul style="list-style-type: none">Ethernet, sender: 01:4D:C9:0A:97:B1 receiver: 01:68:AE:60:AD:C8IP packet, sender: 164.44.38.13, receiver: 164.44.38.12 TTL: 64TCP, sender port: 7777, receiver port: 7777<ul style="list-style-type: none">ISN 2365, ACK 3663flags: SYN, ACKsent 164.44.38.12 >> 164.44.38.13 Type: TCP<ul style="list-style-type: none">Ethernet, sender: 01:68:AE:60:AD:C8 receiver: 01:4D:C9:0A:97:B1IP packet, sender: 164.44.38.12, receiver: 164.44.38.13 TTL: 64TCP, sender port: 7777, receiver port: 7777<ul style="list-style-type: none">ISN 0, ACK 2365flags: Acksent 164.44.38.12 >> 164.44.38.13 Type: TCP<ul style="list-style-type: none">Ethernet, sender: 01:68:AE:60:AD:C8 receiver: 01:4D:C9:0A:97:B1IP packet, sender: 164.44.38.12, receiver: 164.44.38.13 TTL: 64TCP, sender port: 7777, receiver port: 7777<ul style="list-style-type: none">ISN 2365, ACK 0flags: Finreceived 164.44.38.13 >> 164.44.38.12 Type: TCP<ul style="list-style-type: none">Ethernet, sender: 01:4D:C9:0A:97:B1 receiver: 01:68:AE:60:AD:C8IP packet, sender: 164.44.38.13, receiver: 164.44.38.12 TTL: 64TCP, sender port: 7777, receiver port: 7777<ul style="list-style-type: none">ISN 0, ACK 2366flags: Ack	<ul style="list-style-type: none">received 164.44.38.12 search 164.44.38.13 Type: ARP request<ul style="list-style-type: none">Ethernet, sender: 01:68:AE:60:AD:C8 receiver: FF:FF:FF:FF:FF:FF> ARP-request:sent 164.44.38.12 found 164.44.38.13 Type: ARP response<ul style="list-style-type: none">Ethernet, sender: 01:4D:C9:0A:97:B1 receiver: 01:68:AE:60:AD:C8> ARP-response:received 164.44.38.12 >> 164.44.38.13 Type: TCP<ul style="list-style-type: none">Ethernet, sender: 01:68:AE:60:AD:C8 receiver: 01:4D:C9:0A:97:B1IP packet, sender: 164.44.38.12, receiver: 164.44.38.13 TTL: 64TCP, sender port: 7777, receiver port: 7777<ul style="list-style-type: none">ISN 3663, ACK 0flags: SYNsent 164.44.38.13 >> 164.44.38.12 Type: TCP<ul style="list-style-type: none">Ethernet, sender: 01:4D:C9:0A:97:B1 receiver: 01:68:AE:60:AD:C8IP packet, sender: 164.44.38.13, receiver: 164.44.38.12 TTL: 64TCP, sender port: 7777, receiver port: 7777<ul style="list-style-type: none">ISN 2365, ACK 3663flags: SYN, ACKreceived 164.44.38.12 >> 164.44.38.13 Type: TCP<ul style="list-style-type: none">Ethernet, sender: 01:68:AE:60:AD:C8 receiver: 01:4D:C9:0A:97:B1IP packet, sender: 164.44.38.12, receiver: 164.44.38.13 TTL: 64TCP, sender port: 7777, receiver port: 7777<ul style="list-style-type: none">ISN 0, ACK 2365flags: Ackreceived 164.44.38.12 >> 164.44.38.13 Type: TCP<ul style="list-style-type: none">Ethernet, sender: 01:68:AE:60:AD:C8 receiver: 01:4D:C9:0A:97:B1IP packet, sender: 164.44.38.12, receiver: 164.44.38.13 TTL: 64TCP, sender port: 7777, receiver port: 7777<ul style="list-style-type: none">ISN 2365, ACK 0flags: Finsent 164.44.38.13 >> 164.44.38.12 Type: TCP<ul style="list-style-type: none">Ethernet, sender: 01:4D:C9:0A:97:B1 receiver: 01:68:AE:60:AD:C8IP packet, sender: 164.44.38.13, receiver: 164.44.38.12 TTL: 64TCP, sender port: 7777, receiver port: 7777<ul style="list-style-type: none">ISN 0, ACK 2366flags: Ack

Рис.3: Журнал сообщений Computer 1 и Computer 2

Взаимодействие пакетов и кадров, передаваемых по сети при использовании транспортного протокола TCP я захотел рассмотреть подробнее, так как в отличие от UDP ещё его детально не анализировал.

Стоит отметить, что принцип работы концентратора не поменялся, поэтому поведение пакетов через него будет аналогично пункту 3.3 с использованием протокола UDP.

1. ARP-обмен (определение MAC-адресов)

Перед началом TCP-соединения компьютеры должны узнать MAC-адреса друг друга для взаимодействия на канальном уровне. Тут очевидно, уже не раз это описывал.

2. Установление TCP-соединения (трехэтапное рукопожатие)

TCP использует механизм "трехэтапного рукопожатия" (three-way handshake) для установления соединения между узлами. Этот механизм обеспечивает надежное соединение и согласование начальных порядковых номеров.

Первый этап (SYN)

- Computer1 отправляет сегмент с флагом SYN (Synchronize)
 - Отправитель (MAC): 01:68:AE:60:AD
 - Получатель (MAC): 01:4D:C9:0A:97
 - IP-пакет: отправитель 164.44.38.12, получатель 164.44.38.13
 - TCP-сегмент:
 - Порт отправителя: 7777
 - Порт получателя: 7777
 - Порядковый номер (ISN): 3663
 - Флаг: SYN (запрос на синхронизацию)
 - ACK: 0 (не является подтверждением)

Флаг SYN указывает, что компьютер иницирует новое соединение и предлагает свой начальный порядковый номер (ISN) 3663.

Второй этап (SYN-ACK)

- Computer2 отвечает сегментом с флагами SYN и ACK
 - Отправитель (MAC): 01:4D:C9:0A:97
 - Получатель (MAC): 01:68:AE:60:AD
 - IP-пакет: отправитель 164.44.38.13, получатель 164.44.38.12
 - TCP-сегмент:
 - Порт отправителя: 7777
 - Порт получателя: 7777
 - Порядковый номер (ISN): 2365
 - Подтверждение (ACK): $3663+1=3664$
 - Флаги: SYN, ACK

Computer2 подтверждает получение SYN, отправляя флаг ACK со значением 3664 (ISN Computer1 + 1) и устанавливает свой собственный порядковый номер 2365 с помощью флага SYN.

Третий этап (ACK)

- Computer1 подтверждает получение SYN-ACK
 - Отправитель (MAC): 01:68:AE:60:AD
 - Получатель (MAC): 01:4D:C9:0A:97
 - IP-пакет: отправитель 164.44.38.12, получатель 164.44.38.13
 - TCP-сегмент:
 - Порт отправителя: 7777
 - Порт получателя: 7777
 - Порядковый номер: 0 (в данном случае представлен как 0)
 - Подтверждение (ACK): $2365+1=2366$
 - Флаг: ACK

Computer1 подтверждает получение SYN от Computer2, отправляя ACK со значением 2366 (ISN Computer2 + 1). На этом этапе соединение считается установленным и готовым к передаче данных.

3. Передача данных

После установления соединения происходит обмен данными между компьютерами. Каждый сегмент содержит порядковый номер и номер подтверждения для обеспечения надежной доставки.

Передача данных от Computer1 к Computer2

- Computer1 отправляет сегмент с данными
 - Отправитель (MAC): 01:68:AE:60:AD
 - Получатель (MAC): 01:4D:C9:0A:97
 - IP-пакет: отправитель 164.44.38.12, получатель 164.44.38.13
 - TCP-сегмент:
 - Порт отправителя: 7777
 - Порт получателя: 7777
 - Порядковый номер: 2365 (следующий ожидаемый байт)
 - Подтверждение (ACK): 0
 - Флаг: FIN

Этот сегмент содержит данные, отправленные от Computer1 к Computer2. Обратите внимание на флаг FIN, который указывает на то, что Computer1 завершил передачу данных и инициирует закрытие соединения.

Подтверждение получения данных от Computer2

- Computer2 подтверждает получение данных
 - Отправитель (MAC): 01:4D:C9:0A:97
 - Получатель (MAC): 01:68:AE:60:AD
 - IP-пакет: отправитель 164.44.38.13, получатель 164.44.38.12
 - TCP-сегмент:
 - Порт отправителя: 7777
 - Порт получателя: 7777
 - Порядковый номер: 0
 - Подтверждение (ACK): 2366
 - Флаг: ACK

Computer2 подтверждает получение данных и FIN-флага, отправляя ACK со значением 2366.

4. Завершение соединения (четырёхэтапное рукопожатие)

Завершение TCP-соединения происходит через "четырёхэтапное рукопожатие", где каждая сторона должна явно закрыть свою часть соединения.

Первый этап (FIN от Computer1)

- Как уже отмечено выше, Computer1 отправил FIN-флаг, что означает, что он завершил передачу данных

Второй этап (ACK от Computer2)

- Computer2 подтверждает получение FIN-флага путем отправки ACK

Третий этап (FIN от Computer2)

- Computer2 также отправляет FIN-флаг, указывая, что тоже завершил передачу данных
 - Отправитель (MAC): 01:4D:C9:0A:97
 - Получатель (MAC): 01:68:AE:60:AD
 - IP-пакет: отправитель 164.44.38.13, получатель 164.44.38.12
 - TCP-сегмент:
 - Порт отправителя: 7777
 - Порт получателя: 7777
 - Порядковый номер: 2365
 - Подтверждение (ACK): 0
 - Флаг: FIN

Четвертый этап (ACK от Computer1)

- Computer1 отправляет окончательное подтверждение (ACK) в ответ на FIN от Computer2
 - Отправитель (MAC): 01:68:AE:60:AD
 - Получатель (MAC): 01:4D:C9:0A:97
 - IP-пакет: отправитель 164.44.38.12, получатель 164.44.38.13
 - TCP-сегмент:
 - Порт отправителя: 7777
 - Порт получателя: 7777
 - Порядковый номер: 0
 - Подтверждение (ACK): 2366
 - Флаг: ACK

После этого обмена соединение считается полностью закрытым.

4 Этап 2. Локальная сеть с коммутатором

4.1 Построение сети

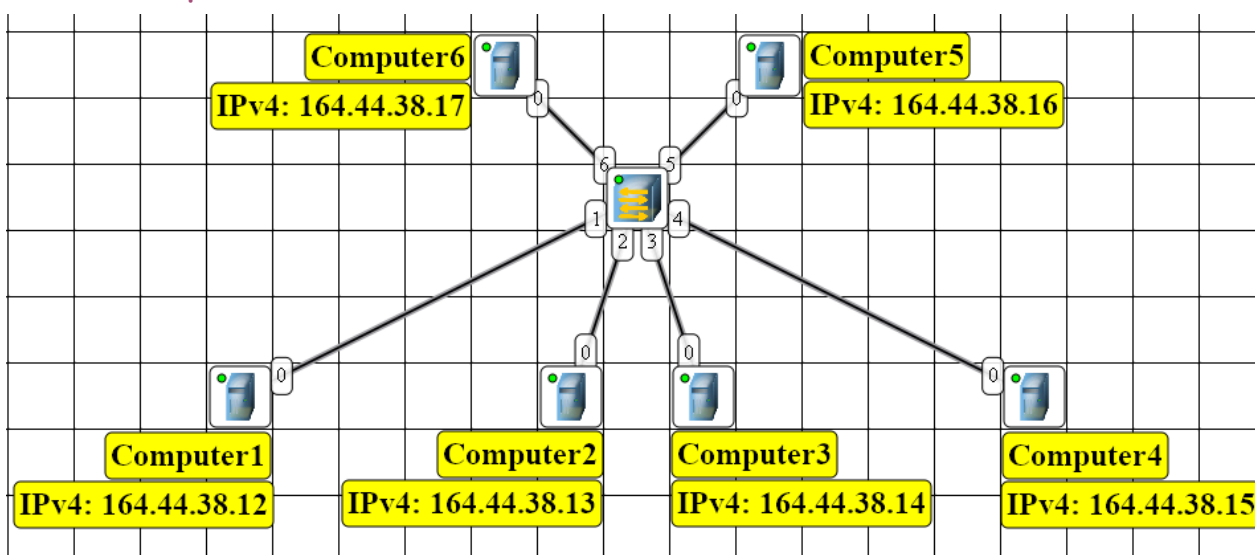


Рис.4: Модель локальной сети из 6 компьютеров с коммутатором

4.2 Анализ таблиц

Будем честны, тут просто нечего анализировать, всё то же самое, причём ещё со времён первой лабораторной.

Отправляя запросы между всеми компьютерами пополняются ARP-таблицы записями о соответствии MAC-адреса и IP-адреса других компьютеров посредством ARP-запросов и ARP-ответов. Единственное отличие с концентратором тут в том, что

коммутатор умный и содержит свою табличку коммутации арг-запросов, которые быстро оттуда пропадают.

	Mac-address	Port	Record type	TTL
1	01:68:AE:60:AD:C8	LAN1	Dinamic	41
2	01:C6:3D:61:87:CF	LAN6	Dinamic	41
3	01:4D:C9:0A:97:B1	LAN2	Dinamic	23
4	01:E3:00:4E:A7:45	LAN4	Dinamic	3

Поля таблицы:

1. MAC-адрес
2. Порт
3. Тип записи
4. TTL (измеряется в секундах, время жизни одной записи = 300)

Заполнение таблицы происходит тогда, когда один из компьютеров отправляет через коммутатор запрос и при этом компьютера-отправителя нет в таблице коммутации. Как только компьютер добавляется в таблицу, начинает отсчитываться время жизни данного соединения.

Например вот я многократно послал данные из различных компьютеров к другим и появилось 4 записи о соответствии мас-адреса и порта, и для них начался отсчёт времени жизни соединения. Где 3 - там только начался. Где 41 - там уже давно идёт.

Таблица коммутации будет построена полностью, если все компьютеры, которые подключены к данному коммутатору хотя бы один запрос за 300 секунд с момента появления в таблице первой записи. Поэтому максимальное количество строк в таблице равняется количеству подключенных к коммутатору компьютеров.

4.3 Тестирование сети (отправка пакетов)

Абсолютно ничего интересного. Аналогично предыдущим пунктам

5 Этап 3. Многосегментная локальная сеть

5.1 Построение сети

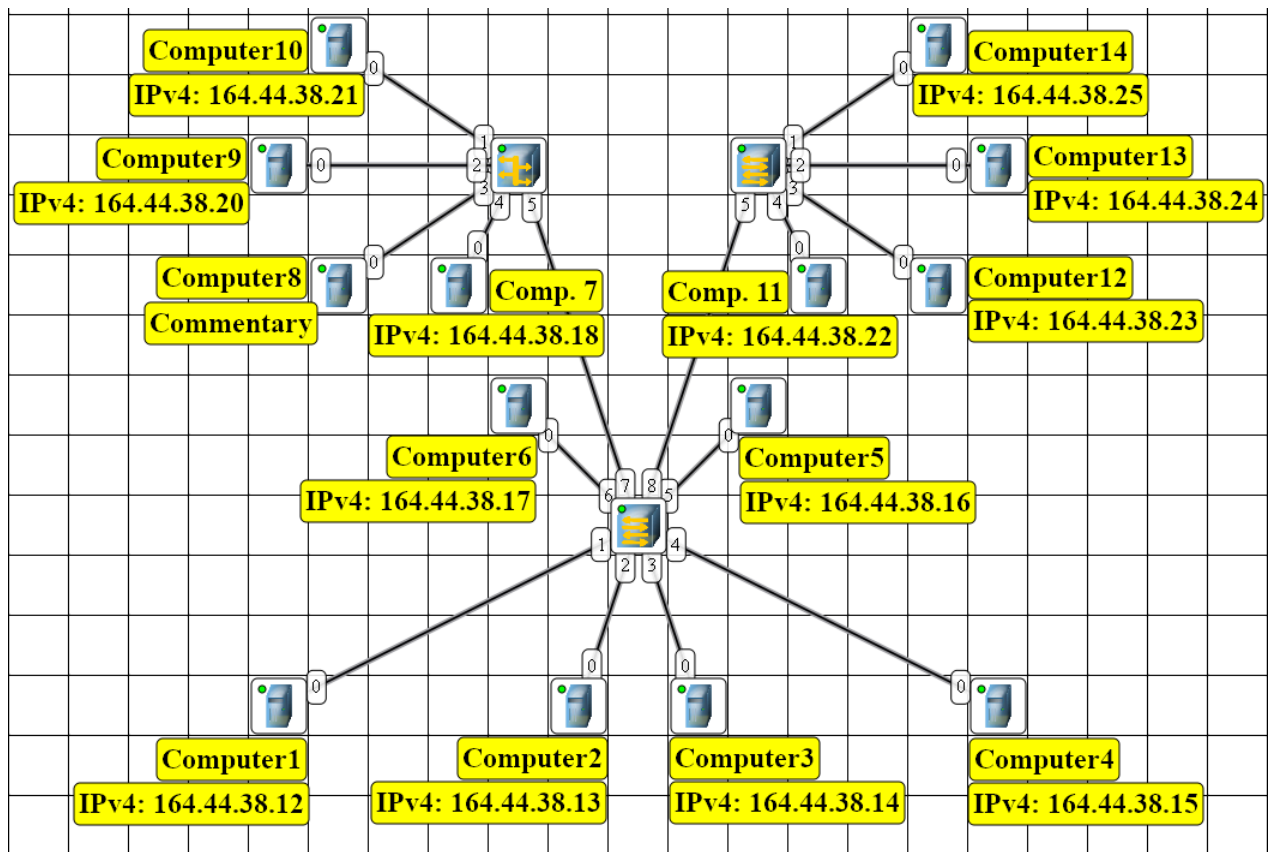


Рис.4: Модель полносвязной сети из 3х компьютеров

5.2 Тестирование сети (отправка пакетов), анализ таблиц

Тут всё скучно и то же самое.

6 Выводы

Самое интересное - почему я именно таким образом соединил сети?

Для начала объясню выбор количества компьютеров - мне просто было скучно, и я хотел симметричную картинку)

Затем вопрос топологии - кольцо тут не сделать, потому что концентратор не может получать и передавать одновременно более одного сообщения, а если поменять его на коммутатор, то есть высокий риск возникновения петель коммутации. Надо настраивать STP и т.п.

Топология общей шины обеспечивает простое соединение всех устройств к единой среде передачи, однако имеет существенные ограничения по масштабируемости и становится неэффективной при увеличении числа узлов из-за роста коллизий.

Поэтому я выбрал простое последовательное подключение, ну или если хотите "звездообразное", потому что чем-то это подключение действительно похоже на топологию звезда. Для полноценной звезды должен был бы быть коммутатор в центре, который будет разгружать все остальные узлы, и одновременно добавлять надежности (и одновременно убавлять), добавлять задержку и т.д.

Так много компьютеров я как раз добавил чтобы проиллюстрировать как сильно может быть перегружен коммутатор (в особенности нижний), чтобы его нужно было разгружать и пользоваться маршрутизаторами (хотя на самом деле 4 обычных компьютера это ничто для современного коммутатора. Проблемы начинаются только при пропускных способностях за Терабайт/сек, нужных при формировании какого-нибудь AI кластера в домашних условиях. И это не шутка, это реалии 2025 года 😊)

Помимо этого скажу, что я погрузился достаточно глубоко в протокол TCP и теперь мне понятен принцип его работы, это было крайне увлекательно и полезно.

Грустно что несмотря на понимание и знания, я думаю что всё равно паршиво напишу компьютерное тестирование, которое весит гораздо больше лабораторных, просто потому что не способен бездумно заучивать информацию. А в тестах спрашивают не принципы, а знания чиселок. Потом когда в иностранной магистратуре будут рассматривать кандидатов - посмотрят в диплом, увидят 4 за предмет компьютерных сетей, и подумают что человек плохо их знает. Хотя 4 это не за знание будет, а за зазубривание чиселок.