

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Основы компьютерных сетей

ОТЧЕТ
по лабораторной работе №6
на тему
ПРОТОКОЛ TCP

Студент

Т.Ю. Петрович

Преподаватель

В.А. Марцинкевич

МИНСК 2024

1 ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

1.1 Исходные данные к работе

Изучить практические особенности использования протокола TCP. Использовать отладочные средства, такие как Wireshark и tcpdump.

Теоретическая часть. Изобразить диаграмму взаимодействия между клиентом (С) и сервером (S) по протоколу TCP с детализацией до флагов SYN, ACK и FIN; полей SN и AN; а также количества данных -- с учетом одного из дополнений по варианту.

№ п/п.	Данные С -> S	Данные С <- S	MSSes	Дополнение
22	2333	20	815	SWS (Clark)

2.1. Взаимодействие изобразить полностью (без сокращений), включая установление соединения, пересылку данных и закрытие соединения.

2.2. Соединение устанавливать по стандарту RFC 793.

2.3. Служебные сегменты изображать пунктирной линией, сегменты с данными -- сплошной.

2.4. Можно использовать кумулятивные подтверждения.

2.5. Указанное дополнение продемонстрировать по крайней мере один раз.

2.6. Если дополнение связано с другими флагами или полями, то указать значения этих флагов или полей в отношении каждого сегмента.

2.7. Применительно ко всем дополнениям, кроме «разупорядочивание», данные передавать по одному сегменту («маятником»).

2.8. Применительно к дополнениям «SWS (Nagle)», «SWS (Clark)», привести дополнительные пояснения о том, как они реализованы в данном случае.

2.9. Применительно ко всем дополнениям, кроме «>1 сегмента одновременно» и «разупорядочивание», данные передавать по одному сегменту («маятником»).

3.0. Практическая часть.

3.1) запустить *Wireshark*;

3.2) запустить процесс захвата трафика;

3.3) открыть любой сайт; в URL должны присутствовать любые три буквы из фамилии студента в латинской транскрипции;

3.4) остановить захват трафика;

3.5) настроить фильтр просмотра для просмотра сегментов tcp.

Пояснить каким образом происходит начало TCP-сессии (процесс трехэтапного рукопожатия)

2 ОБЗОР ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ

2.1 Теоретическая часть

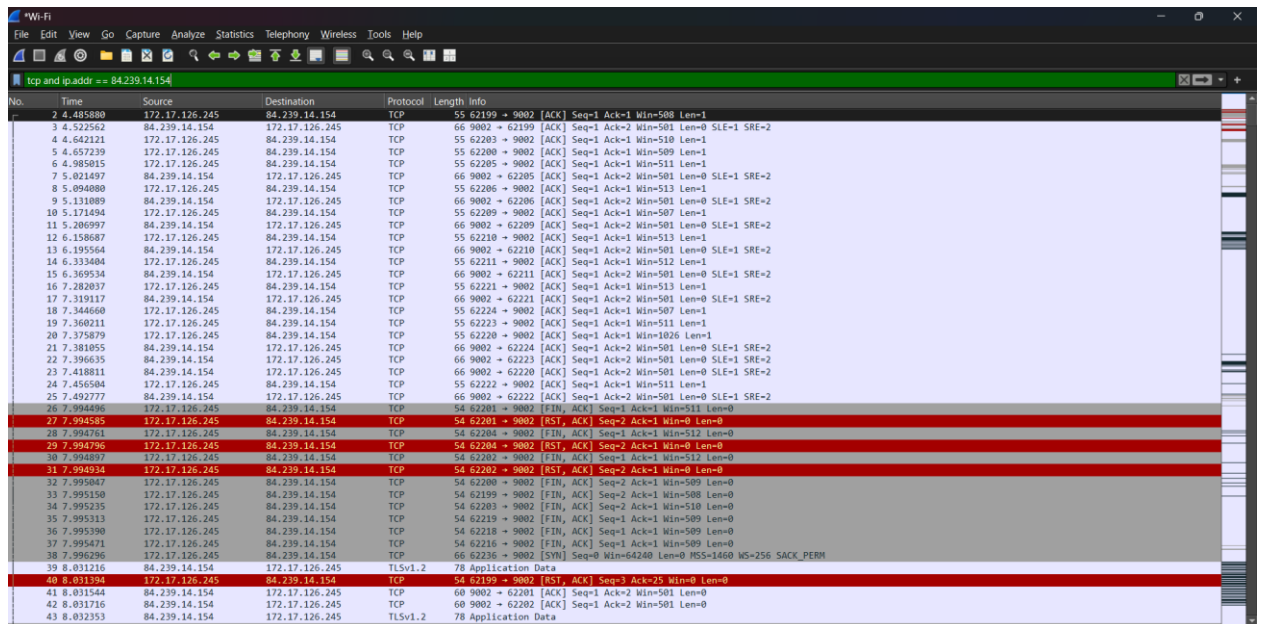
Синдром глупого окна (SWS) — это проблема в компьютерных сетях, вызванная плохо реализованным управлением потоком TCP. Серьезная проблема в работе скользящего окна может возникнуть, когда отправляющая прикладная программа медленно создает данные, принимающая прикладная программа медленно потребляет данные или и то, и другое. Если сервер с этой проблемой не может обработать все входящие данные, он просит своих клиентов уменьшить объем данных, которые они отправляют за один раз (настройка окна для TCP-пакета). Если сервер по-прежнему не в состоянии обработать все входящие данные, окно становится все меньше и меньше, иногда до такой степени, что передаваемые данные меньше заголовка пакета, что делает передачу данных крайне неэффективной. Название этой проблемы связано с тем, что размер окна уменьшается до «глупого» значения.

Поскольку с обработкой каждого пакета связаны определенные накладные расходы, увеличение числа пакетов означает увеличение накладных расходов на обработку уменьшающегося объема данных. Конечный результат – взбучивание.

Когда приемник создает синдром глупого окна, используется решение Дэвида Д. Кларка. Решение Кларка закрывает окно до тех пор, пока не будет получен еще один сегмент максимального размера сегмента (MSS) или буфер не будет наполовину пуст.

4 РЕЗУЛЬТАТЫ РАБОТЫ

При выполнении практической части задания был захвачен и проанализирован трафик, при подключении к сайту <https://www.yahoo.com> Результат работы представлен на рисунке 4.1.



No.	Time	Source	Destination	Protocol	Length	Info
2	4.485880	172.17.126.245	84.239.14.154	TCP	55	62199 → 9002 [ACK] Seq=1 Ack=1 Win=508 Len=1
3	4.522562	84.239.14.154	172.17.126.245	TCP	66	9002 → 62199 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
4	4.642121	172.17.126.245	84.239.14.154	TCP	55	62203 → 9002 [ACK] Seq=1 Ack=1 Win=510 Len=1
5	4.657239	172.17.126.245	84.239.14.154	TCP	55	62200 → 9002 [ACK] Seq=1 Ack=1 Win=509 Len=1
6	4.985015	172.17.126.245	84.239.14.154	TCP	55	62205 → 9002 [ACK] Seq=1 Ack=1 Win=511 Len=1
7	5.021497	84.239.14.154	172.17.126.245	TCP	66	9002 → 62205 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
8	5.094080	172.17.126.245	84.239.14.154	TCP	55	62206 → 9002 [ACK] Seq=1 Ack=1 Win=513 Len=1
9	5.131089	84.239.14.154	172.17.126.245	TCP	66	9002 → 62206 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
10	5.171494	172.17.126.245	84.239.14.154	TCP	55	62209 → 9002 [ACK] Seq=1 Ack=1 Win=507 Len=1
11	5.206997	84.239.14.154	172.17.126.245	TCP	66	9002 → 62209 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
12	6.158087	172.17.126.245	84.239.14.154	TCP	55	62210 → 9002 [ACK] Seq=1 Ack=1 Win=513 Len=1
13	6.195564	84.239.14.154	172.17.126.245	TCP	66	9002 → 62210 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
14	6.333404	172.17.126.245	84.239.14.154	TCP	55	62211 → 9002 [ACK] Seq=1 Ack=1 Win=512 Len=1
15	6.369534	84.239.14.154	172.17.126.245	TCP	66	9002 → 62211 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
16	7.282037	172.17.126.245	84.239.14.154	TCP	55	62221 → 9002 [ACK] Seq=1 Ack=1 Win=513 Len=1
17	7.319117	84.239.14.154	172.17.126.245	TCP	66	9002 → 62221 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
18	7.344660	172.17.126.245	84.239.14.154	TCP	55	62224 → 9002 [ACK] Seq=1 Ack=1 Win=507 Len=1
19	7.360211	172.17.126.245	84.239.14.154	TCP	55	62223 → 9002 [ACK] Seq=1 Ack=1 Win=511 Len=1
20	7.375879	172.17.126.245	84.239.14.154	TCP	55	62220 → 9002 [ACK] Seq=1 Ack=1 Win=1026 Len=1
21	7.381055	84.239.14.154	172.17.126.245	TCP	66	9002 → 62224 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
22	7.396635	84.239.14.154	172.17.126.245	TCP	66	9002 → 62223 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
23	7.418811	84.239.14.154	172.17.126.245	TCP	66	9002 → 62220 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
24	7.456584	172.17.126.245	84.239.14.154	TCP	55	62222 → 9002 [ACK] Seq=1 Ack=1 Win=511 Len=1
25	7.492777	84.239.14.154	172.17.126.245	TCP	66	9002 → 62222 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
26	7.994456	172.17.126.245	84.239.14.154	TCP	54	62201 → 9002 [FIN, ACK] Seq=1 Ack=1 Win=511 Len=0
27	7.994585	172.17.126.245	84.239.14.154	TCP	54	62201 → 9002 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
28	7.994761	172.17.126.245	84.239.14.154	TCP	54	62204 → 9002 [FIN, ACK] Seq=1 Ack=1 Win=512 Len=0
29	7.994756	172.17.126.245	84.239.14.154	TCP	54	62204 → 9002 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
30	7.994857	172.17.126.245	84.239.14.154	TCP	54	62202 → 9002 [FIN, ACK] Seq=1 Ack=1 Win=512 Len=0
31	7.994934	172.17.126.245	84.239.14.154	TCP	54	62202 → 9002 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
32	7.995047	172.17.126.245	84.239.14.154	TCP	54	62200 → 9002 [FIN, ACK] Seq=2 Ack=1 Win=509 Len=0
33	7.995150	172.17.126.245	84.239.14.154	TCP	54	62199 → 9002 [FIN, ACK] Seq=2 Ack=1 Win=508 Len=0
34	7.995235	172.17.126.245	84.239.14.154	TCP	54	62203 → 9002 [FIN, ACK] Seq=2 Ack=1 Win=510 Len=0
35	7.995313	172.17.126.245	84.239.14.154	TCP	54	62219 → 9002 [FIN, ACK] Seq=1 Ack=1 Win=509 Len=0
36	7.995390	172.17.126.245	84.239.14.154	TCP	54	62218 → 9002 [FIN, ACK] Seq=1 Ack=1 Win=509 Len=0
37	7.995471	172.17.126.245	84.239.14.154	TCP	54	62216 → 9002 [FIN, ACK] Seq=1 Ack=1 Win=509 Len=0
38	7.996206	172.17.126.245	84.239.14.154	TCP	66	62236 → 9002 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
39	8.031216	84.239.14.154	172.17.126.245	TLSv1.2	78	Application Data
40	8.031394	172.17.126.245	84.239.14.154	TCP	54	62199 → 9002 [RST, ACK] Seq=3 Ack=25 Win=0 Len=0
41	8.031544	84.239.14.154	172.17.126.245	TCP	60	9002 → 62201 [ACK] Seq=1 Ack=2 Win=501 Len=0
42	8.031716	84.239.14.154	172.17.126.245	TCP	60	9002 → 62202 [ACK] Seq=1 Ack=2 Win=501 Len=0
43	8.032353	84.239.14.154	172.17.126.245	TLSv1.2	78	Application Data

Рисунок 4.1 – Процесс захвата трафика с помощью программы wireshark

5 ВЫВОДЫ

Изучение практических особенностей использования протокола TCP с применением отладочных средств, таких как Wireshark и tcpdump, является важной задачей для понимания работы этого протокола на низком уровне. Анализ взаимодействия между клиентом (C) и сервером (S) по протоколу TCP с детализацией до флагов SYN, ACK и FIN, полей SN и AN, а также количества передаваемых данных, позволяет глубже понять механизмы, обеспечивающие надежную и упорядоченную передачу данных.

Выполнение данной практической работы позволит получить глубокое понимание принципов работы протокола TCP и его практического применения с использованием современных отладочных средств.