

Практика 7. Транспортный уровень. UDP (сдать до 09.04.2022)

1. Wireshark: UDP (5 баллов)

Начните захват пакетов в приложении Wireshark и затем сделайте так, чтобы ваш хост отправил и получил несколько UDP-пакетов (например, с помощью обращений DNS).

Выберите один из UDP-пакетов и разверните поля UDP в окне деталей заголовка пакета.

Ответьте на вопросы ниже, представив соответствующие скрины программы Wireshark.

1. Выберите один UDP-пакет. По этому пакету определите, сколько полей содержит UDP-заголовок.
2. Определите длину (в байтах) для каждого поля UDP-заголовка, обращаясь к отображаемой информации о содержимом полей в данном пакете.
3. Значение в поле Length (Длина) – это длина чего?
4. Какое максимальное количество байт может быть включено в полезную нагрузку UDP-пакета?
5. Чему равно максимально возможное значение номера порта отправителя?
6. Какой номер протокола для протокола UDP? Дайте ответ и для шестнадцатеричной и десятичной системы. Чтобы ответить на этот вопрос, вам необходимо заглянуть в поле Протокол в IP-дейтаграмме, содержащей UDP-сегмент.
7. Проверьте UDP-пакет и ответный UDP-пакет, отправляемый вашим хостом. Определите отношение между номерами портов в двух пакетах.

2. Программирование. Эхо запросы через UDP

Реализуйте *сервер* для пингования, а также его *клиента*. Сделайте скриншоты, подтверждающие корректную работу вашей программы пингования со стороны клиента.

А. Серверная часть (2 балла)

Сервер находится в бесконечном цикле, ожидая приходящие UDP-пакеты.

Если пакет прибывает, то сервер просто изменяет символы входящего сообщения на заглавные и отправляет их обратно клиенту.

Серверный код должен моделировать 20% потерю пакетов.

Б. Клиентская часть (2 балла)

Клиент должен отправить 10 эхо-запросов серверу. Поскольку UDP является ненадежным с точки зрения доставки протоколом, то пакет, отправленный от клиента к серверу или наоборот, может быть потерян в сети. Так как клиент не может бесконечно ждать ответа на запрос, нужно задать период ожидания ответа (тайм-аут), равный одной секунде – если ответ не будет получен в течение одной секунды, клиентская программа должна предполагать, что пакет потерян.

Ваша клиентская программа должна:

- отправить эхо-запрос, используя UDP
- распечатать ответное сообщение от сервера (если такое есть)
- вычислить и вывести на печать время оборота (RTT) в секундах для каждого пакета при ответе сервера
- в противном случае, вывести сообщение «Request timed out»

Формат сообщения:

Ping *номер_последовательности* *время*

- номер_последовательности начинается с 1 и увеличивается до 10 для каждого последующего сообщения, отправленного клиентом
- время – это момент времени, когда клиент отправляет сообщение

В. Вывод в формате ring (2 балла)

Версия клиента из предыдущей части (Б) вычисляет время оборота для каждого пакета и выводит его отдельно. Измените вывод таким образом, чтобы он соответствовал тому, как это делается в стандартной утилите ping.

Для этого вам нужно будет сообщить минимальное, максимальное и среднее значение RTT в конце каждого ответа от сервера. Дополнительно вычислите коэффициент потери пакетов (в процентах).

Г. UDP Heartbeat (4 балла)

UDP Heartbeat (монитор доступности) подобен программе пингования. Он может быть использован для проверки, работает ли приложение, и вывода сообщения об односторонней потере пакетов.

Клиент отправляет порядковый номер и текущую временную метку в пакете UDP на сервер, который слушает «сердцебиение» (т.е. ожидает UDP-пакеты) клиента. После получения пакетов сервер вычисляет разницу во времени и сообщает о потерях. Если пакеты отсутствуют определенный период времени, заданный параметром, то делается предположение, что клиентское приложение остановлено и соответствующее сообщение выводится на консоль сервера.

Реализуйте UDP Heartbeat (обе части – клиент и сервер), доработав обе ваши части программы пингования из заданий А и Б.

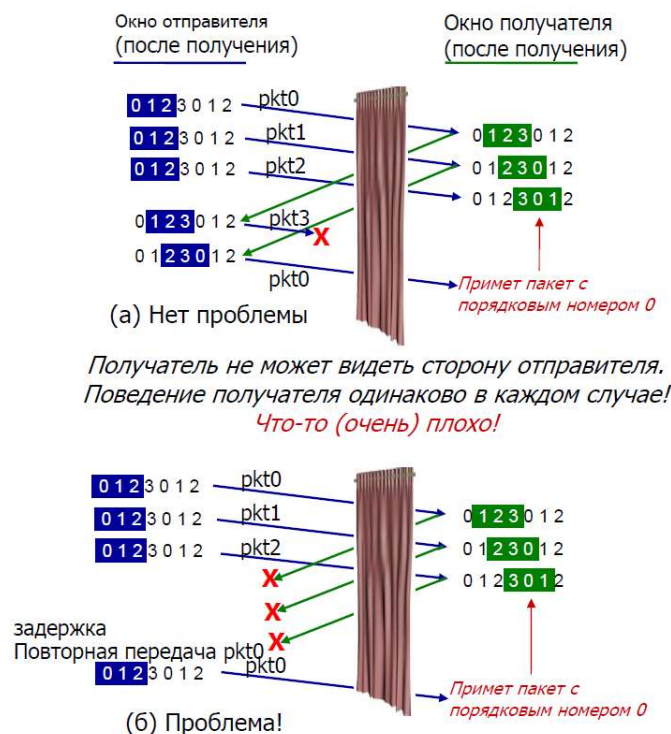
Обратите внимание, что клиентов у сервера может быть сразу несколько одновременно. Протестируйте такой сценарий.

3. Задачки

Задача 1 (3 балла)

Рассмотрим протоколы GBN и SR. Предположим, пространство порядковых номеров имеет размер k .

Постановка задачи: найти наибольшее допустимое окно передачи, которое позволит обойти возникновение проблемы, показанной на рисунке для каждого из этих протоколов?



Описание проблемы:

Отсутствие синхронизации между окнами отправителя и получателя имеет важные последствия, когда мы сталкиваемся с ограниченностью диапазона порядковых номеров. Рассмотрим, что могло бы произойти, например, если у нас есть четыре пакета с порядковыми номерами 0, 1, 2, 3, а размер окна равен трем. Предположим, пакеты с 0 по 2 переданы отправителем, корректно получены и подтверждены получателем. В этот момент окно получателя заполняется четвертым, пятым и шестым пакетами, которые имеют порядковые номера 3, 0 и 1, соответственно. Теперь рассмотрим два сценария.

В первом сценарии (на рисунке сверху) квитанции ACK на первые три пакета доставлены верно. Таким образом, отправитель сдвигает окно вперед и отправляет четвертый, пятый и шестой пакеты с порядковыми номерами 3, 0 и 1 соответственно.

Пакет с порядковым номером 3 потерян, но пакет с порядковым номером 0, **содержащий новые данные**, получен.

Во втором случае (рисунок снизу) ACK пакеты для первых трех пакетов данных потеряны, и отправитель пересылает эти пакеты. Таким образом, получатель далее получает пакет с порядковым номером 0 — **копию первого отправленного**.

Теперь рассмотрим ту же ситуацию с точки зрения принимающей стороны. Действия, выполняемые передающей стороной, скрыты от нее; принимающая сторона способна лишь следить за последовательностями получаемых пакетов и генерируемых квитанций. Подобная ограниченность приводит к тому, что обе описанные выше ситуации воспринимаются принимающей стороной как одинаковые. Она не может отличить исходную передачу пакета от повторной.

Задача 2 (2 балла)

Представим себе следующую ситуацию: один хост расположен в Санкт-Петербурге, а другой — во Владивостоке. Пусть время оборота RTT между этими двумя хостами приблизительно равно 30 мс. Предположим далее, что хосты соединены каналом со скоростью передачи R , равной 1 Гбит/с (10^9 бит/с).

Предположим, что размер передаваемого пакета составляет 1500 байт, включая поля заголовка и данные.

Насколько большим должен быть размер окна n , чтобы использование канала составило более 98 процентов?