

Санкт-Петербургский политехнический университет Петра Великого
Высшая школа прикладной математики и вычислительной физики
Кафедра прикладной математики

Лабораторная работа
по дисциплине «Компьютерные сети»
на тему

**Реализация протоколов автоматического запроса
повторной передачи Go-Back-N и Selective Repeat**

Выполнил
студент группы 5040102/00201
Руководитель
доцент, к.ф.-м.н.

Н.В. Суханов

А.Н. Баженов

Санкт-Петербург
2022

Постановка задачи

Реализовать систему, состоящую из отправителя (Sender) и получателя (Receiver), способных обмениваться сообщениями по каналу связи через протоколы автоматического запроса повторной передачи Go-Back-N (GBN) и Selective Repeat (SRP) [1]. Канал связи может допускать потерю пакетов с заданной вероятностью.

Реализация

Протоколы передачи данных реализованы Python. Sender и Receiver работают в двух отдельных потоках, создаваемых с использованием модуля threading. Важной особенностью модуля является то, он из-за глобальной блокировки интерпретатора (GIL) не поддерживает истинного параллелизма, то есть, потоки исполняются конкурентно, а не параллельно.

В обоих протоколах передача данных осуществляется пакетами, содержащими уникальный индекс, непосредственно передаваемые данные, а также служебный код, который используется для передачи сообщений об ошибках и остановке работы.

Каждый протокол поддерживает два параметра: window_size – размер скользящего окна (количество последовательных сообщений, которое Sender может отправить без подтверждения от Receiver'a о получении одного из сообщений) и timeout – время (в секундах) ожидания Sender'ом подтверждения от Receiver' о получении сообщения, после чего сообщение считается потерянным и требует повторной отправки.

При тестировании во всех тестах передавался массив данных из 50 элементов. После завершения передачи производилась проверка, что данные переданы полностью и без искажений.

Код проекта выложен на GitHub:

<https://github.com/NikitaSukhanov/ComputerNetworks/tree/lab1>

Результаты

Рассмотрим, как зависит количество сообщений, которые потребовались протоколам, чтобы передать данные, и общее время передачи (в секундах) от параметров протоколов и вероятности потери сообщения в канале.

Для начала убедимся, что при нулевой вероятности потери сообщения, оба протокола передают ровно по 50 сообщений, независимо от других параметров (Рис. 1, Рис. 2).

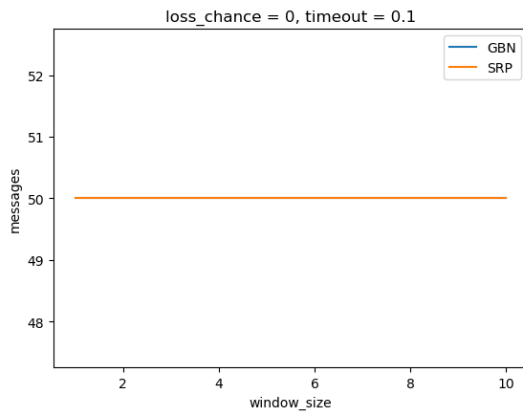


Рис. 1 Зависимость количества сообщений от размера окна без потери сообщений

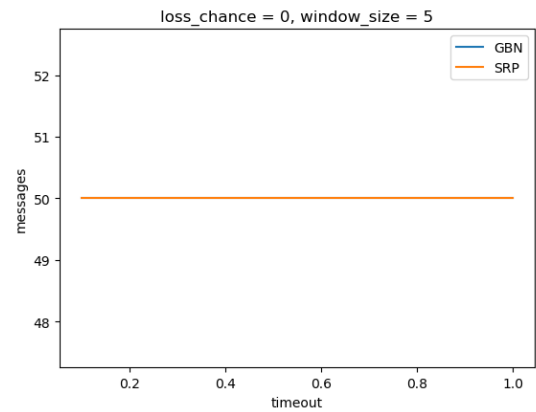


Рис. 2 Зависимость количества сообщений от времени ожидания подтверждения без потери сообщений

Также рассмотрим время работы протоколов при тех же параметрах. На Рис. 3 можно увидеть, что время передачи данных для обоих протоколов монотонно убывает при увеличении размера скользящего окна. Скорее всего, в данном случае это является особенностью системы, в которой моделируется передача. Так как потоки работают по очереди, при увеличении размера окна уменьшается количество необходимых переключений между потоками. Например, при единичном размере окна, интерпретатору необходимо 50 раз передать управление между потоками Sender'а и Receiver'а, в то время как при размере окна равном 10, переключений требуется всего 5, так как сначала Sender может передать 10 сообщений подряд, а потом Receiver может вернуть ему сразу 10 подтверждений. Также рассмотрим зависимость от времени ожидания подтверждения (таймаута) (Рис. 4). Так как потери сообщений не происходит, то при достаточно больших значениях таймаута, время работы от него почти не зависит. Однако стоит отметить, что если выбрать значение таймаута слишком маленьким, то подтверждения от Receiver'а перестанут успевать доходить до Sender'а, что может привести к повторной отправке доставленных пакетов, и как следствие, к увеличению времени работы.

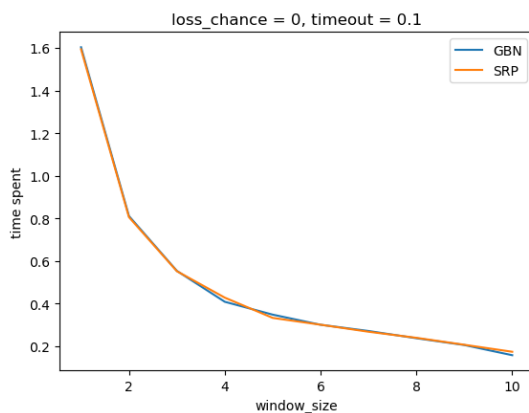


Рис. 3 Зависимость времени работы от размера окна без потери сообщений

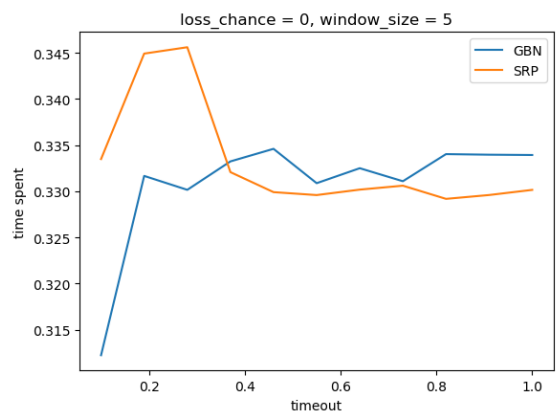


Рис. 4 Зависимость времени работы от времени ожидания подтверждения без потери сообщений

Теперь проведём аналогичные тесты, но с большой (90%) вероятностью потери сообщений в канале. Из Рис. 5 можно заметить, что количество передаваемых сообщений для протокола GBN растёт с увеличением размера окна, в то время как для SRP остаётся почти постоянным. Это связано с тем, что в тестируемой реализации GBN в случае не получения подтверждения, или получения непоследовательного подтверждения, протокол повторяет отправку всех сообщений из скользящего окна, а в протоколе SRP сообщения могут подтверждаться непоследовательно, а решение о повторной отправке принимается для каждого сообщения индивидуально. Также стоит отметить (Рис. 5, Рис. 6), что количество передаваемых сообщений у протокола GBN в целом выше, чем у SRP.

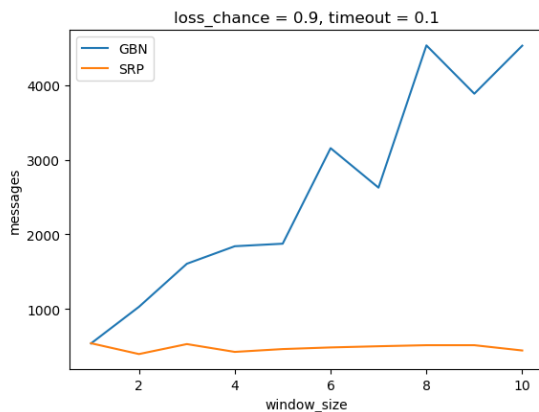


Рис. 5 Зависимость количества сообщений от размера окна с потерей сообщений

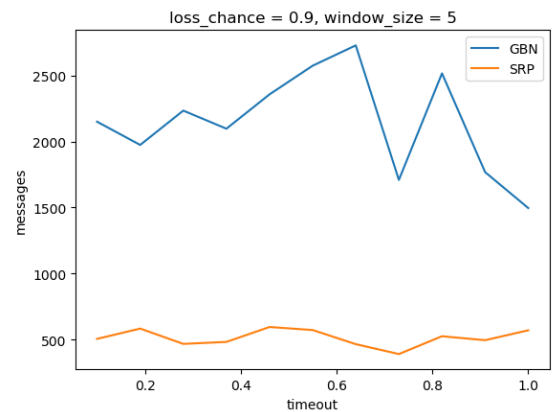


Рис. 6 Зависимость количества сообщений от времени ожидания подтверждения с потерей сообщений

На Рис. 7 представлена зависимость времени работы от размера окна. Для протокола SRP время работы монотонно убывает с ростом размера окна, в то время как как с протоколом GBN ситуация не столь очевидна. Можно заметить интересную закономерность, что несмотря на то, что количество отправленных протоколом GBN сообщений существенно увеличивается с ростом размера окна, время, требуемое ему на передачу данных, при этом не растёт.

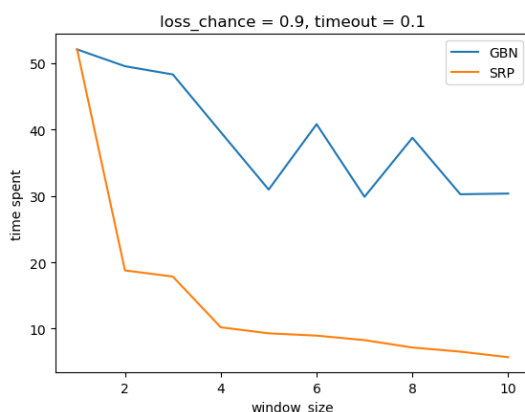


Рис. 7 Зависимость времени работы от размера окна с потерей сообщений

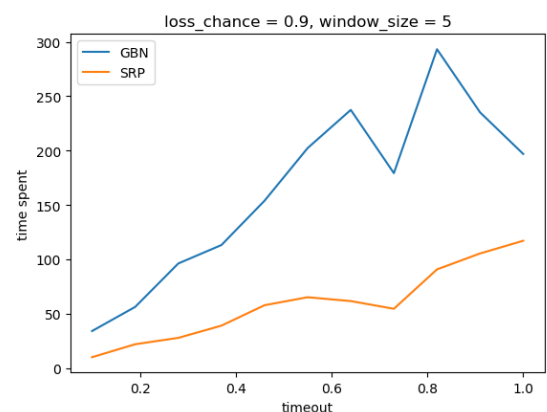


Рис. 8 Зависимость времени работы от времени ожидания подтверждения с потерей сообщений

Для обоих протоколов время работы возрастает с увеличением таймаута (Рис. 8). Это можно объяснить тем, что большинство пакетов не доходит до получателя, и таким образом при снижении задержки снижается время, бездействия Sender'a, когда он уже отправил все пакеты, и ожидает подтверждения от Receiver'a о доставке пакета, который скорее всего потерялся. Тем не менее, здесь также неэффективно выставить таймаут слишком маленьким, иначе Sender будет повторно отправлять даже доставленные сообщения из-за того, что до него не успело дойти подтверждение.

Выводы

В рамках работы были реализованы протоколы автоматического запроса повторной передачи данных Go-Back-N и Selective Repeat. При тестировании была исследована зависимость метрик эффективности протоколов, таких как количество передаваемых сообщений и время работы от параметров протоколов и вероятности потери сообщения в канале. Стоит учитывать, что некоторые из приведённых закономерностей являются свойствами самих протоколов, в то время как другие – особенностями среды тестирования, связанными с конкурентным исполнением.

Использованная литература

1. А.Н. Баженов, Компьютерные сети, курс лекций