



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»

---

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

---

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## *К КУРСОВОМУ ПРОЕКТУ*

### *НА ТЕМУ:*

*Реализация межсетевого экрана*

Студент ИУ7-72Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата) Е.В. Брянская  
(И.О.Фамилия)

Руководитель курсового проекта

\_\_\_\_\_  
(Подпись, дата) Н.Ю. Рязанова  
(И.О.Фамилия)

2021 г.



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ  
Заведующий кафедрой ИУ7  
(Индекс)  
И.В. Рудаков  
(И.О.Фамилия)  
« \_\_\_\_ » \_\_\_\_\_ 2021 г.

**ЗАДАНИЕ**  
**на выполнение курсового проекта**

по дисциплине Компьютерные сети

Студенты группы ИУ7-72Б

Брянская Екатерина Вадимовна  
(Фамилия, имя, отчество)

Иванов Всеволод Алексеевич  
(Фамилия, имя, отчество)

Тема курсового проекта BitTorrent-клиент

Направленность КП (учебный, исследовательский, практический, производственный, др.)  
учебный

Источник тематики (кафедра, предприятие, НИР) кафедра

График выполнения проекта: 25% к 4 нед., 50% к 7 нед., 75% к 11 нед., 100% к 14 нед.

**Задание** Разработать торрент-клиент на основе протокола BitTorrent

**Оформление курсового проекта:**

Расчетно-пояснительная записка на 20-30 листах формата А4.

Расчетно-пояснительная записка должна содержать постановку задачи, введение,  
аналитическую, конструкторскую, технологическую части, заключение, список литературы.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.):  
на защиту работы должна быть предоставлена презентация, состоящая из 15-20 слайдов.

На слайдах должны быть отражены: постановка задачи, использованные методы и  
алгоритмы, расчетные соотношения, структура комплекса программ, интерфейсы.

Дата выдачи задания «8» октября 2021 г.

Руководитель курсового проекта

Н.О. Рогозин  
(Подпись, дата) (И.О.Фамилия)

Студент

Е.В. Брянская  
(Подпись, дата) (И.О.Фамилия)

Студент

В.А. Иванов  
(Подпись, дата) (И.О.Фамилия)

## Содержание

<b>ВВЕДЕНИЕ</b>	<b>4</b>
<b>1 Аналитическая часть</b>	<b>5</b>
1.1 Постановка задачи . . . . .	5
1.2 Принцип работы протокола . . . . .	5
1.3 Структура .torrent файла . . . . .	6
1.4 Взаимодействие клиента и сервера . . . . .	7
1.5 Структура сообщений . . . . .	8
1.6 Взаимодействие клиентов . . . . .	8
<b>2 Технологическая часть</b>	<b>9</b>
2.1 Выбор технологических средств . . . . .	9
2.2 UML диаграмма классов . . . . .	9
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>12</b>

## ВВЕДЕНИЕ

За последние время существенно возросли объёмы информации, передаваемой по сети Интернет. Очевидно, что подобная тенденция сохранится и в будущем – будет расти число пользователей и объём потребляемого ими трафика.

В подобных условиях актуальным является вопрос производительности серверов. Ввиду описанных выше факторов нагрузка на них будет постоянно расти, что будет вынуждать их владельцев производить их обновление и расширение или снижение скорости обмена информацией с клиентами.

Последнее является чувствительным для загрузки файлов больших объёмов. Решением в таком случае может быть кооперативный обмен файлами. Наиболее популярным протоколом для этой технологии является Bittorrent.

**Целью** данной работы является разработка Bittorrent клиента.

Для достижения поставленной цели необходимо решить следующие **задачи**:

- 1) изучить структуру и принцип работы протокола;
- 2) разработать алгоритм взаимодействия с сервером и клиентами;
- 3) реализовать программу для загрузки файлов на основе протокола Bittorrent.

# 1 Аналитическая часть

## 1.1 Постановка задачи

Результатом работы должна стать программа для загрузки файлов по протоколу Bittorrent, удовлетворяющая следующим требованиям:

- поддерживать файлы расширения torrent;
- поддерживать функцию загрузки данных как от сервера, так и от других клиентов;
- обладать графическим интерфейсом для удобства выполнения действий и просмотра текущей информации по состоянию загрузки.

Первостепенной задачей для дальнейшей разработки является изучения устройства выбранного протокола.

## 1.2 Принцип работы протокола

Bittorrent – P2P протокол для кооперативного обмена файлами через интернет.

В данном протоколе выделены две роли:

- 1) **пир** (клиент) хранит файлы и производит обмен их частями с другими пирами;
- 2) **трекер** (сервер) хранит таблицу файлов и список пиров, имеющих данный файл в распоряжении.

Пир, желающий получить файл должен обладать **.torrent файлом**, с помощью которого он может обратиться к серверу. Сервер предоставляет адреса клиентов, обладающих запрашиваемыми файлами после чего начинается их загрузка. Передача осуществляется частями (**pieces**), каждый torrent-клиент, скачивая эти части, в то же время отдаёт их другим клиентам, что снижает нагрузку на каждого отдельного клиента.

### 1.3 Структура .torrent файла

Как было отмечено выше, первым шагом в начале загрузки является получение и парсинг файла специального расширения .torrent.

Для кодирования данных в .torrent-файлах используется формат Bencode. Само содержимое – ассоциативный массив с полями:

- **info** – вложенный ассоциативный массив который собственно и описывает файлы, которые передаёт торрент;
- **announce** – URL трекера;
- **announce-list** – список трекеров, если их несколько, в Bencode-виде — список списков;
- **creation date** – дата создания;
- **comment** – текстовое описание торрента;
- **created by** – автор торрента.

info и announce являются обязательными полями, всё остальное — опционально. Первый в свою очередь состоит из:

- **piece length** – размер одного куска;
- **pieces** – конкатенация SHA1-хешей каждого куска (каждый хэш - 20 символов);
- **name** – имя файла (если файл один);
- **length** – содержит длину файла (если файл один);
- **files** – если файлов несколько, то содержит список ассоциативных массивов (с указанием length и path).

Данная информация используется на всём протяжении загрузки файла и его последующей раздаче.

## 1.4 Взаимодействие клиента и сервера

Чтобы перейти к загрузке файла клиент должен получить список пиров у трекера. Для этого он должен отправить GET-запрос, называемый анонсом, по адресу announce по пути /announce. После данного действия трекер узнаёт о наличии нового клиента, и может выдать его адрес другим клиентам. Указываются следующие URL-параметры.

- **info\_hash** – SHA1-хеш словаря info. Используется для поиска файла в таблице трекера, то есть фактически является его уникальным идентификатором.
- **peer\_id** – уникальный ID клиента. Имеет вид *-<2-символьный id><номер версии из 4 цифр>-<12 случайных цифр>*. Такой код может быть сгенерирован клиентом самостоятельно, так как вероятность коллизии с другими клиентами крайне мала (число возможных вариантов peer\_id одной версии превышает количество IPv4 адресов более чем в 200 раз).
- **uploaded, downloaded, left** – количество отправленных, загруженных и незагруженных байтов.
- **port** – TCP-порт, прослушиваемый клиентом. Общепринятыми значениями являются 6881-6889.
- **compact** – принимает ли клиент компактный список пиров.

В случае, если запрос прошёл успешно и по info\_hash был найден необходимый torrent, трекер посылает ответ (также по протоколу HTTP). В его теле содержится следующие поля в формате Bencode:

- **interval** – интервал в секундах до того, как клиент должен сделать новый запрос к трекеру;
- **peers** – список пиров. В случае, если в запросе compact был равен 1, в ответе будет список будет заменён бинарной строкой, которую потребуется разбить на группы по 6 байт для выделения IPv4 адреса и порта каждого

пира.

Подобные запросы будут повторяться раз в interval секунд для поддержания сервера в курсе актуального состояния загрузки и для получения новых адресов пиров.

## 1.5 Структура сообщений

Протокол BitTorrent определяет следующий способ обмена сообщениями для клиентов, его особенности:

- использует стек TCP/IP;
- файл передаётся по кускам фиксированного размера, не в порядке их следования в файле.

Определена следующая структура p2p сообщения:

- 1) **длина**, Len (4Б) – размер типа и полезной нагрузки сообщения;
- 2) **тип**, ID (1Б) – определяет вид сообщения и способ его обработки;
- 3) **полезная нагрузка**, Payload (0 - 32КБ) - содержит передаваемую информацию.

## 1.6 Взаимодействие клиентов



## **2 Технологическая часть**

### **2.1 Выбор технологических средств**

### **2.2 UML диаграмма классов**

На Рисунке 2.1 приведена UML-диаграмма классов.

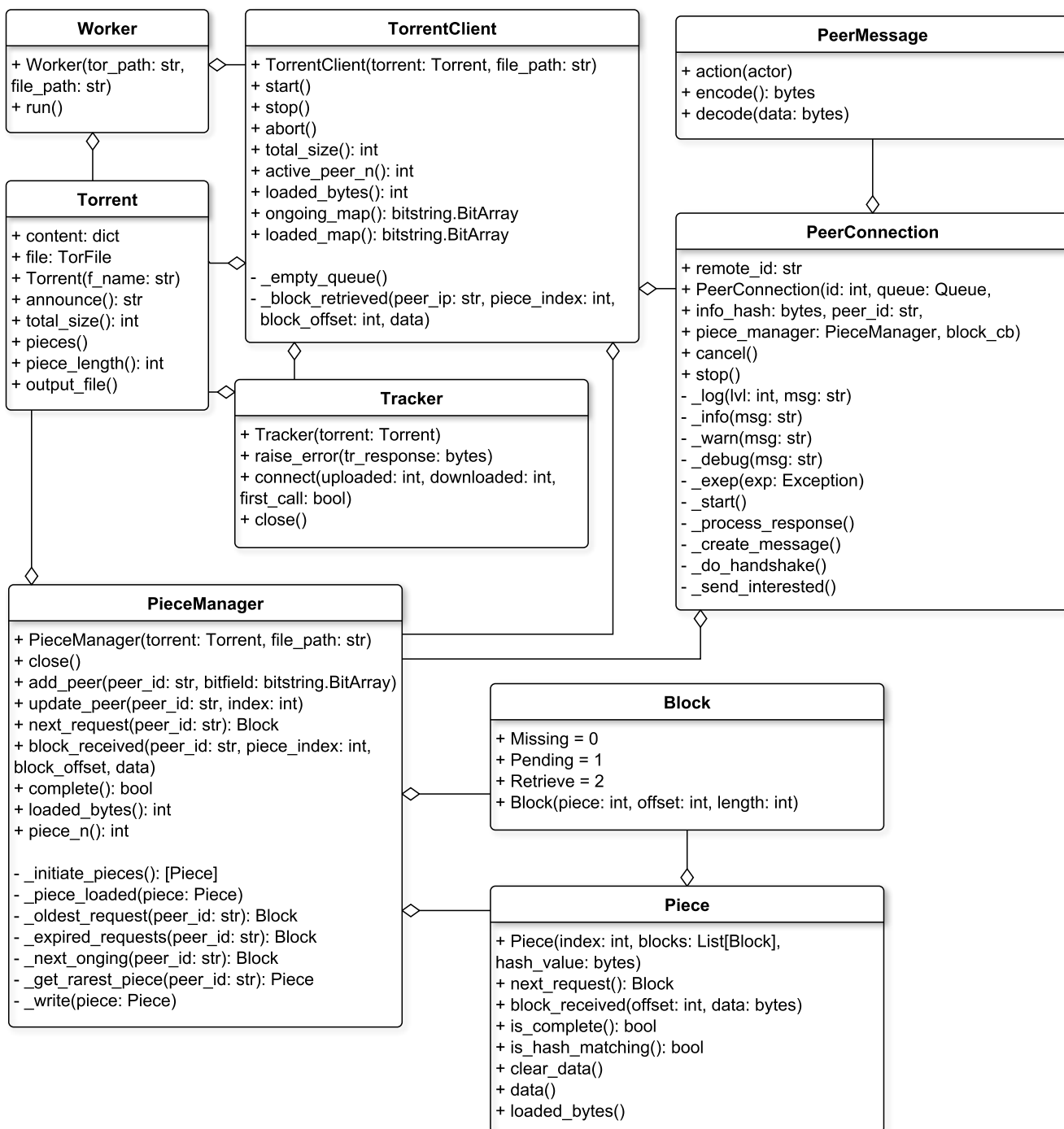


Рисунок 2.1 — \*\*\*\*

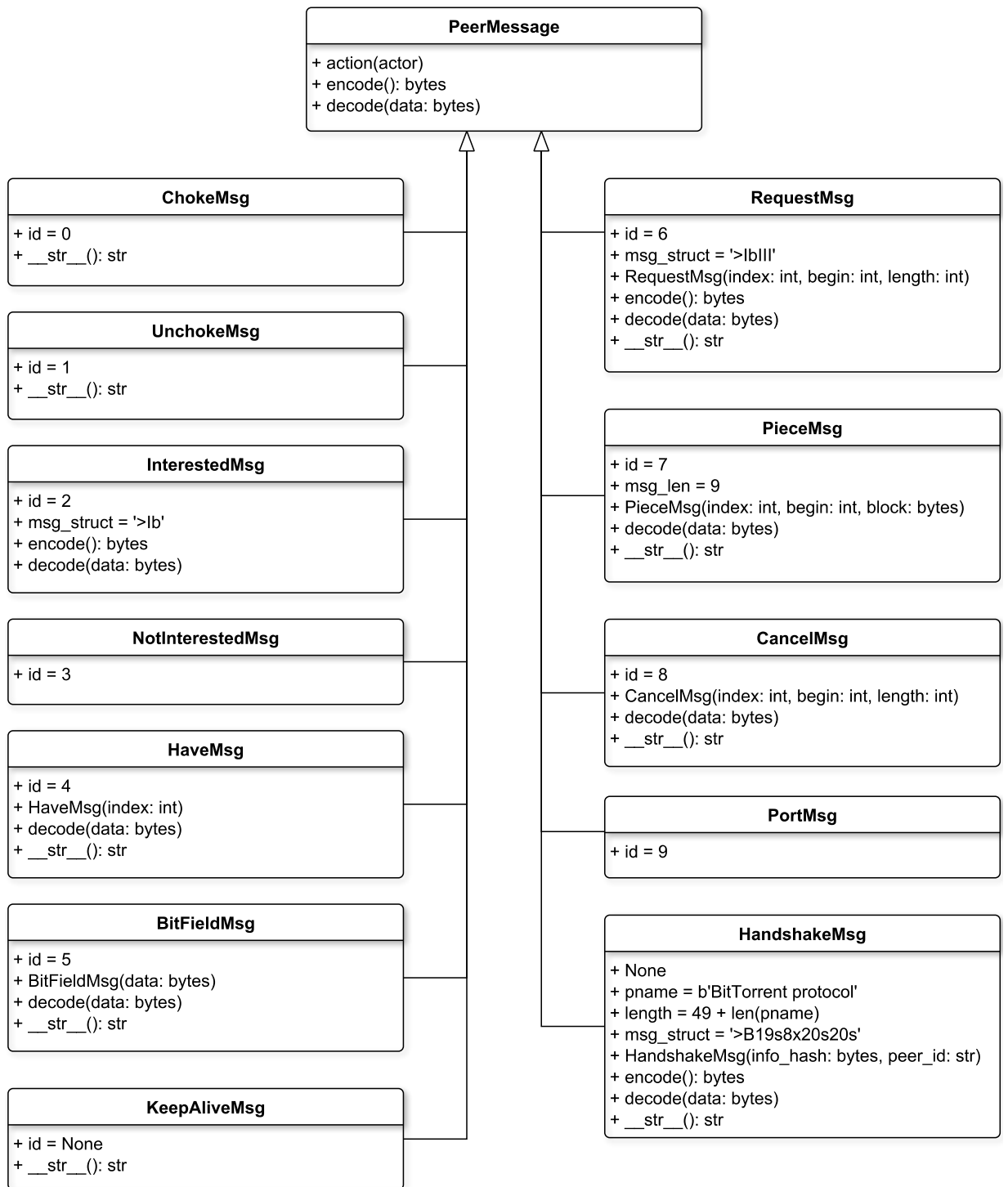


Рисунок 2.2 – \*\*\*\*

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

### **Список литературы**

1. Hari Balakrishnan, M. FransKaashoek , David Karger, Robert Morris, and Ion Stoica. Looking up DATA in P2P systems. In Proc. Acm SIGCOMM'01, San Diego, CA, Aug. 2001.