

ИССЛЕДОВАНИЕ ЗАВИСИМОСТИ ВРЕМЕНИ ПЕРЕДАЧИ ДАННЫХ ПО СЕТИ ОТ ИСПОЛЬЗУЕМЫХ ПРОТОКОЛОВ

Стеклов В.В., Абросимов Л.И.

В настоящее время огромное количество устройств общаются между собой, используя сеть передачи данных. Участники сетевого взаимодействия обязаны согласовать друг с другом множество правил, которые объединяются в протоколы передачи данных. Протоколы нижних уровней сетевой модели OSI часто реализуются с использованием программных и аппаратных средств.[1] Все протоколы, начиная с транспортного уровня и выше, реализуются программными средствами.[2]

В наше время очень актуальна проблема быстрой и надежной передачи информации. Повысить эффективность транспорта данных достаточно сложно. В зависимости от используемых транспортных протоколов время и качество передаваемой информации может сильно отличаться.

Таким образом, в научной работе определяются зависимости времени передачи данных от используемых протоколов транспортного и прикладного уровней. Эффективность передачи информации по сети во многом зависит от её рода. В исследовательской работе осуществляется передача графических изображений между двумя компьютерами, находящимися в одной локальной сети.

К разрабатываемой модели передачи данных определяются качественные оценки обслуживания. Чаще всего параметры, фигурирующие в разнообразных определениях качества обслуживания, регламентируют следующие показатели работы сети:[1]

- скорость передачи данных;
- задержки передачи пакетов;
- уровень потерь и искажений пакетов.

Относительно этих оценок осуществляется выбор наилучшего рассмотренного метода передачи информации.

Простейшая модель передачи данных разрабатывается на языке программирования C# с использованием технологий библиотек .NET. В работе также исследуется роль сжатия графической информации для её эффективной передачи по каналам связи.

Литература

1. **Олифер В. Г., Олифер Н. А.** Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 4-е изд. — СПб.: Питер, 2010. — 944 с.: ил.
2. **Руденков Н.А., Долинер Л.И.** Основы сетевых технологий: Учебник для вузов. Екатеринбург: Изд-во Уральского. Федерального ун-та, 2011. — 300 с.

Национальный исследовательский
университет
МЭИ

**Исследование зависимости
времени передачи данных по
сети от используемых протоколов**

Автор: Стеклов Владислав Витальевич, А-07м-20

Научный руководитель: Абросимов Леонид

Иванович, доктор технических наук, профессор.

Цели исследовательской работы

➤ В зависимости от используемых протоколов время и качество передаваемой информации может сильно отличаться

Протоколы передачи данных, используемые в прикладных программах, основываются на известных протоколах транспорта TCP и UDP

➤ Исследование зависимостей времени передачи данных по сети от использованных протоколов

Данные могут передаваться по различным типам линий – от витой пары до оптоволокна и спутниковых каналов. Но правила транспорта данных и установки логического соединения будут всегда одними и теми же



Актуальность быстрой и надежной передачи информации

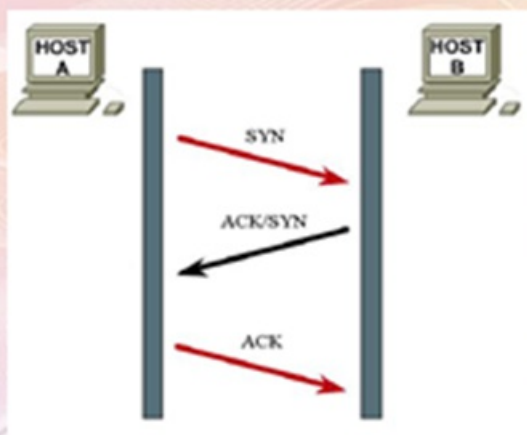
- Распространение Big Data, цифровые снимки
- Географически распределенные команды
- Телеком-СМИ и развлечения
- Обмен файлами. Копирование, синхронизация. Передача миллионов файлов и массивов данных. Поточковая передача данных



TCP

Абстракция надежного сетевого соединения

- повторная передача потерянных данных – требуется дополнительное процессорное время на обработку подтверждений
- доставка данных в определенном порядке – клиент блокируется до получения очередного подтверждения
- установка соединения – проблема долгой установки соединения

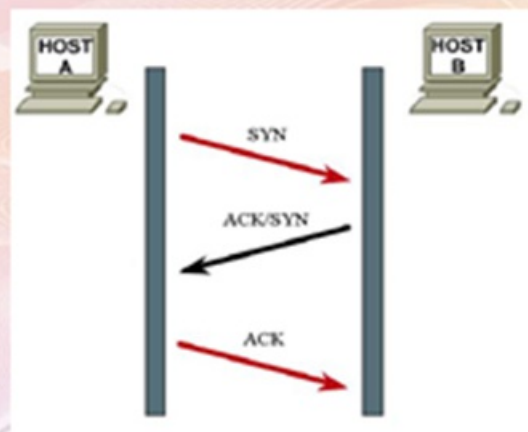


логическое соединение между прикладными процессами

TCP

Абстракция надежного сетевого соединения

- повторная передача потерянных данных – требуется дополнительное процессорное время на обработку подтверждений
- доставка данных в определенном порядке – клиент блокируется до получения очередного подтверждения
- установка соединения – проблема долгой установки соединения



логическое соединение между прикладными процессами

Проблемы TCP

- Медленный старт TCP применяется к каждому новому соединению
- Механизмы контроля потока и перегрузки TCP регулируют пропускную способность всех соединений
- Пропускная способность TCP регулируется через размер окна перегрузки



АЛЬТЕРНАТИВЫ

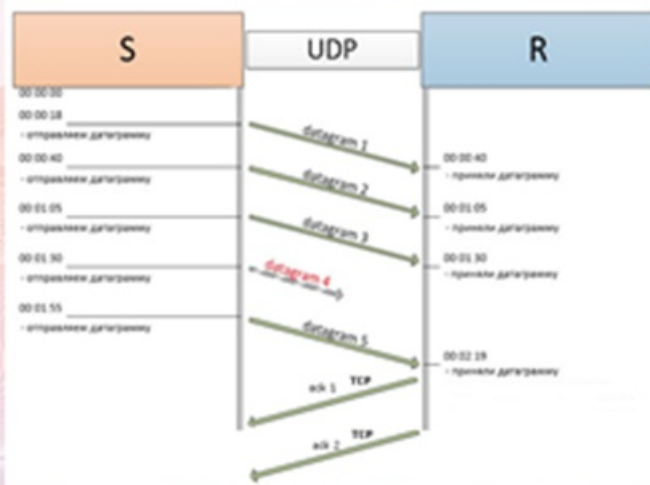
UDP – нет гарантии доставки передаваемой информации

Кэширование (Data Caching) – однотипные данные, несколько потоков

Компрессия, сжатие (Data Compression) – много информации передается в уже сжатом виде

UDP

- Ненадежный
- Нет установки соединения
- Отсутствуют подтверждения
- Нет последовательного приема данных
- Отсутствует окно



Подходит для реализации одноранговой сети

Программный интерфейс отличается от TCP

Практические результаты

Тип файла	Размер файла (байты)	Тип подключения	Время клиента (мс)	Время сервера (мс)
Текстовый документ в формате txt	3 605	TCP	6	12
Текстовый документ в формате txt	3 605	UDP	3	7
Текстовый документ в формате doc	147 773	TCP	22	30
Текстовый документ в формате doc	147 773	UDP	6	-
Изображение png	621 004	TCP	45	48
Изображение png	621 004	UDP	20	-
Изображение jpg	755 637	TCP	51	60
Изображение jpg	755 637	UDP	33	-
Изображение jpg	1 102 805	TCP	58	65
Изображение jpg	1 102 805	UDP	32	-

Переданные изображения



TCP



UDP

Надежный UDP

- Требуется дополнительные методы, позволяющие достигать целостности передаваемых данных
- UDP - протокол с дополнительным уровнем надежности на прикладном уровне

Если какой-то пакет теряется при передаче, не нужно ждать подтверждение о том, получен он или потерян, можно приступать сразу к передаче следующего пакета, а потерянный пересылается в процессе



UDP

```
namespace WebProtocolsModel
{
    [Serializable]
    public class ConfirmMessage
    {
        public int BatchOrder { get; set; }
    }
}
```

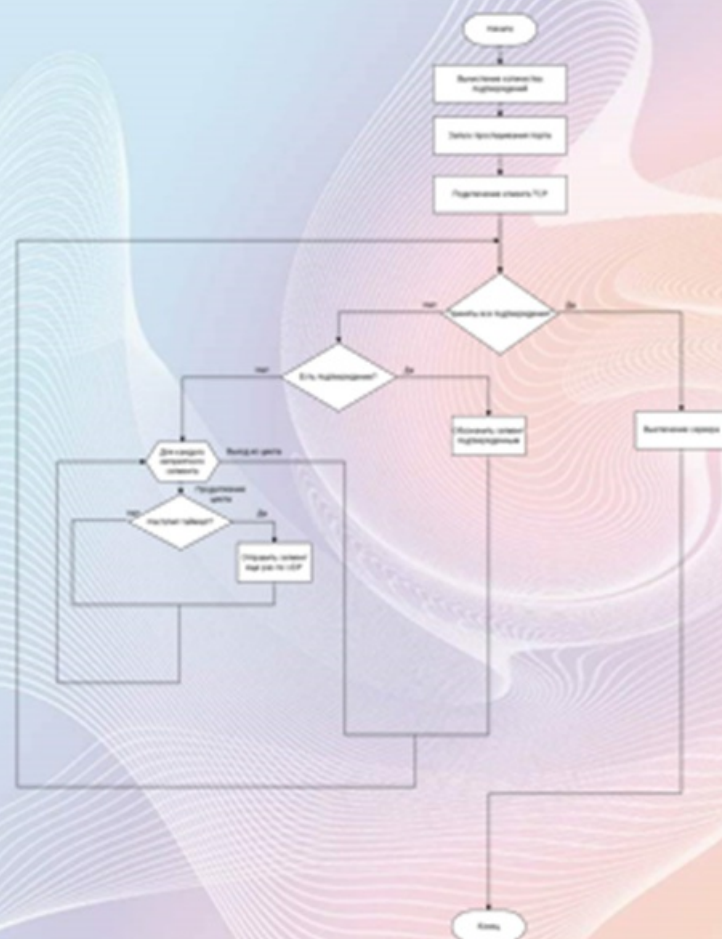
3. Передача подтверждений фактов приемки сообщений сервером осуществляется по протоколу TCP ?

4. Возможность изменять явным образом параметры протокола: размер буферов, время таймаута факта подтверждения принятия данных

1. Надежная доставка пакетов – выходное изображение без искажений
2. Передача изображений осуществляется поверх UDP

```
namespace WebProtocolsModel
{
    [Serializable]
    public class FileBatch
    {
        public byte[] Bytes { get; set; }
        public int Order { get; set; }
    }
}
```

Клиент надежного UDP



- 1. Прием подтверждений от сервера по протоколу TCP
- 2. Обработка подтверждений с помощью механизма таймера
- 3. Отправка потерянных сегментов заново по протоколу UDP.

Таблица 1.
Разный размер буфера при фиксированной длине изображения 1 102 805 байт

Количество сегментов в изображении / Количество повторных отправок	Размер буфера (байты)	Время клиента (мс)	Время сервера (мс)
2154/4055	512	39949	39978
1077/2171	1024	28051	28082
539/625	2048	16674	16698
270/169	4096	11079	11069
135/97	8192	10500	10528

Практические результаты

Таблица 2. Разный размер изображений

Тип файла	Размер файла (байты)	Время клиента (мс)	Время сервера (мс)
Изображение jpg	147 773	10193	10198
Изображение jpg	621 004	26800	26813
Изображение jpg	755 637	31421	31500
Изображение jpg	1 102 805	39949	39978

Таблица 3. Время таймаута при фиксированном изображении 1 102 805 байт и буфере 2К

Время таймаута (сек)	Время клиента (мс)	Время сервера (мс)
1	4629	4619
2	9498	9200
5	16674	16698
8	28543	28557
10	31608	31115

Улучшение показателей

- Время таймера сильно влияет на общее время передачи
- Размер буфера влияет на количество повторных отправок
- Улучшение конфигурации протокола

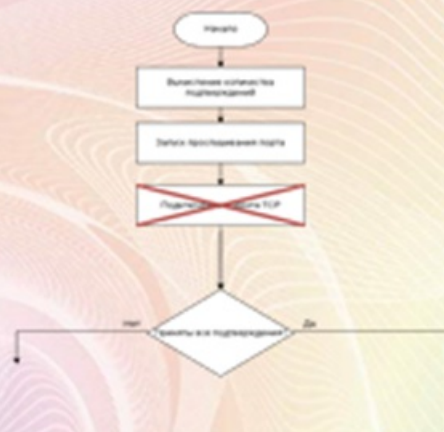


**Практические результаты. Буфер: 8КБ
Таймаут:**

Кол-во ретранс/батчей	Размер файла (байты)	Время клиента (мс)	Время сервера (мс)
30/19	147 773	629	570
145/76	621 004	635	602
209/93	755 637	678	635
149/135	1 102 805	620	649

- Генерация лишних подтверждений от клиента серверу
- Долгая установка соединения
- Общие проблемы TCP

Улучшение показателей



**Практические результаты. Буфер: 8КБ
Таймаут:**

Кол-во ретранс/батчей	Размер файла (байты)	Время клиента (мс)	Время сервера (мс)
10/19	147 773	96	84
54/76	621 004	147	145
117/93	755 637	330	346
123/135	1 102 805	517	440

Роль сжатия информации

Потоковые сокеты – установка соединения, более высокоуровневый интерфейс

- Сжатие информации – уменьшение избыточности данных
- Алгоритмы с потерями (jpg) и без (DEFLATE и GZIP)
- GZIP -> DEFLATE -> LZ77 -> алгоритм Хаффмана
- Повторяющаяся цепочка данных заменяется «ссылками» на её первый экземпляр.
- Кодирование переменной длины



Практические результаты

Таблица 5. Влияние сжатия информации

Тип файла	Размер файла / Сжатый файл (байты)	Тип сжатия	Время клиента (мс)	Время сервера (мс)
Изображение jpg	147 773 / 131072	DEFLATE	48	59
Изображение jpg	147 773 / 131082	GZIP	28	31
Изображение jpg	621 004 / 581632	DEFLATE	98	140
Изображение jpg	621 004 / 581642	GZIP	70	72
Изображение jpg	1 102 805 / 1073152	DEFLATE	144	178
Изображение jpg	1 102 805 / 1073162	GZIP	119	125

Степень сжатия для каждого метода будет определена как: $K_c = V_c / V_o * 100\%$

K_c – коэффициент сжатия, V_c – объем сжатого файла, V_o – исходный объем файла

