

## Лабораторная работа 1

Вычислить эффективность передачи информации пакетами (IP+TCP+данные) -  $N_i/N_0 \cdot (i/(i+s))$ , где  $N_i$ -общее количество правильно принятых битов информационной части пакета,  $N_0$ -общее количество переданных битов.

### Входные параметры:

1.  $i$  - размер информационной части пакета (несколько значений 2,4,8,...,2<sup>16</sup>);
2.  $p$  - вероятность искажения одного бита при передаче (несколько значений 0.01, 0.001, ..., 0.000001);
3.  $s$  - размер служебной части ( $s=20 \cdot 8 + 20 \cdot 8$ );

### Построить, используя oosalc, двумерную диаграмму:

1. По оси абсцисс - длина информационной части кадра,
2. По оси ординат функция: эффективность.
3. Обеспечить возможность задавать вероятность искажения бита.

## Лабораторная работа 2

1. Написать клиент-серверную программу на основе транспортного протокола UDP. Реализовать: подтверждение приема для каждой датаграммы, сохранение целостности всей информации. Клиент передает файл или сообщение (несколькими датаграммами), сервер принимает.
2. Продемонстрировать реализованные возможности программ согласно заданию, при одновременной передаче файлов от нескольких клиентов к серверу. Например, при запуске сервера указать - какие пакеты и сколько раз будут потеряны. Результат правильности приема выводить на экран.
3. Реализация - на языке C/C++, консольные приложения.

Серверная программа должна находить номер свободного порта и выводить его на экран. При запуске клиентской программы задавать со строки IP адрес сервера и порт (например, см. [1], стр. 338-342, функции bind, getsockname).

### Полезные ссылки:

1. Фейт С. TCP/IP: Архитектура, протоколы, реализация (включая IP версии 6 и IP Security). – М.: Лори, 2000. – 424 с.

## Лабораторная работа 3

1. Написать программу обеспечивающую параллельную работу сервера, принимающего файлы от клиента по сети. Условие: мультипроцессная организация на основе функции fork, транспортный протокол – TCP.
2. Обеспечить в сервере завершение «зомби-процессов».
3. Написать клиентскую программу, передающую файл по сети серверу.
4. Продемонстрировать реализованные возможности программ согласно заданию.
5. Серверная программа должна находить номер свободного порта и выводить его на экран. При запуске клиентской программы задавать со строки IP адрес сервера и порт.
6. Реализация - на языке C/C++, консольные приложения.

Серверная программа должна находить номер свободного порта и выводить его на экран. При запуске клиентской программы задавать со строки IP адрес сервера и порт.

## Лабораторная работа 4

1. Написать программу обеспечивающую параллельную работу сервера, принимающего файлы от клиента по сети. Информацию получаемую от клиента сохранять в одном общем файле (обеспечить целостность данных). Условие: мультипоточная организация на основе функций библиотеки pthread, транспортный протокол – TCP.
2. Написать клиентскую программу, передающую файл по сети серверу.
3. Продемонстрировать реализованные возможности программ согласно заданию.
4. Реализация - на языке C/C++, консольные приложения.

Серверная программа должна находить номер свободного порта и выводить его на экран. При запуске клиентской программы задавать со строки IP адрес сервера и порт.

## Лабораторная работа 5

1. Разработать программу однопоточного сервера, использующую асинхронный ввод/вывод (организованный с помощью системного вызова select) обеспечивающую псевдопараллельную работу клиентов.
2. Написать клиентскую программу, передающую сообщения на сервер.
3. Продемонстрировать асинхронную работу сервера. Например, при запуске клиента пользователь задает число  $i$  от 1 до 10. Клиент передает серверу в цикле это число с задержкой в  $i$  секунд между передачей. Сервер отображает на экран полученную от клиентов информацию.

Например:

1-й клиент посылает число 1 в цикле с задержкой в 1 сек.

2-ой клиент посылает число 2 с задержкой в 2 сек.

3-й клиент посылает число 3 в цикле с задержкой в 3 сек.

Сервер отображает информацию полученную от клиентов. Если у Вас правильно организован асинхронный ввод/вывод, то на экран со стороны сервера будет выводиться с чередованием числа 1, 2 и 3. Причем частота появления определенного числа будет зависеть от задержки по времени его передачи.

4. Реализация - на языке C/C++, консольные приложения.

Серверная программа должна находить номер свободного порта и выводить его на экран. При запуске клиентской программы задавать со строки IP адрес сервера и порт (например, см. [1], стр. 338-342, функции bind, getsockname).

### Полезные ссылки:

1. Фейт С. TCP/IP: Архитектура, протоколы, реализация (включая IP версии 6 и IP Security). – М.: Лори, 2000. – 424 с.

## Лабораторная работа 6

1. Написать программу, реализующую работу сервера по двум протоколам (TCP и UDP) параллельно с помощью системного вызова select.
2. Написать две клиентские программы, передающие на сервер файлы по протоколам TCP и UDP соответственно.
3. Реализация - на языке C/C++, консольные приложения.

Серверная программа должна находить номер свободного порта и выводить его на экран. При запуске клиентской программы задавать со строки IP адрес сервера и порт.