

# Математические модели и центральные алгоритмы информационно-вычислительной системы компьютерного тестирования

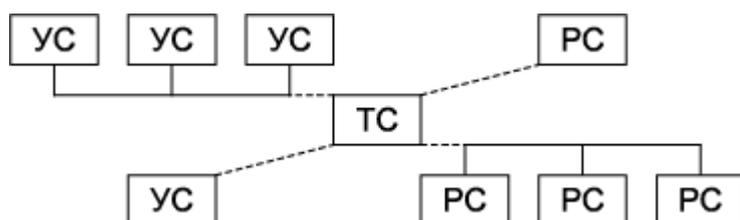
**Нардюжев Виктор Иванович** — доцент кафедры информационно-компьютерных дисциплин Обнинского института атомной энергетики, кандидат технических наук. E-mail: vin@kaluga.ru

**Нардюжев Иван Викторович** — доцент кафедры автоматизированных систем управления Обнинского института атомной энергетики, кандидат технических наук. E-mail: vin@kaluga.ru

## Концептуальная модель ИВС КТ

Пользователей ИВС КТ можно разделить на организаторов тестирования (авторов тестовых заданий, сотрудников центров тестирования, заказчиков), которые с управляющих станций (УС) получают доступ к основным функциям системы, и участников (которые с рабочих станций (РС) непосредственно взаимодействуют только с подсистемами регистрации) проведения сеансов тестирования на компьютерах и передачи данных по каналам связи. В качестве посредника выступает тест-сервер (ТС).

### Структура ИВС КТ



Часто роль УС, ТС и РС выполняют разные компьютеры. Это позволяет использовать оборудование, соответствующее решаемым задачам (например, сервер без мультимедийных средств и рабочие станции с жёсткими дисками небольшой ёмкости), и облегчает реализацию режима секретности. Однако при этом возникает вопрос о создании эффективной подсистемы передачи данных по каналам связи, особенно при проведении сеансов тестирования через Интернет.

В большинстве случаев предполагается единственный ТС, что значительно упрощает контроль и управление, но может стать проблемой при недостаточной производительности и надёжности центрального компьютера. Самым трудным для тест-сервера является вариант работы через Интернет из-за большого числа одновременно обслуживаемых сеансов тестирования и значительного объёма постоянно меняющихся баз данных.

Таким образом, для проверки способности ИВС КТ проводить сеансы тестирования на автономных компьютерах, в локальной сети, через Интернет достаточно оценить её работоспособность в последнем наиболее тяжёлом режиме, приняв в качестве показателя эффективности среднее время отклика на запрос РС при ограничениях на производительность ТС и подсистемы передачи данных по каналам связи.

С этой целью разработаны и исследованы аналитическая модель блока обработки заявок, поступающих на тест-сервер, и имитационная модель телекоммуникационной подсистемы. Исходные данные для моделирования были получены в ходе описанного ниже эксперимента, который проводился для выявления причинно-следственных связей, присущих ИВС КТ и существенных для данного исследования.

В рамках эксперимента на основе анализа информации о зарубежных системах компьютерного тестирования и имеющегося отечественного опыта автоматизации централизованного тестирования на бланках был разработан прототип ИВС КТ.

### Функциональная схема ИВС КТ



Управляющие станции получали доступ к тест-серверу по локальной сети (Ethernet, 10 Мбит/с) для записи набора компьютерных тестов, чтения базы с реквизитами и результатами участников тестирования, обновления итоговой статистической информации о тестируемых.

Тест-сервер через Интернет (на последней миле использовалась аппаратура ADSL, 265 Кбит/с) предоставлял компьютерные тесты рабочим станциям, в автоматическом режиме обрабатывал полученные от них ответы тестируемых и подсчитывал набранные баллы, формируя базу с реквизитами и результатами участников тестирования.

Взаимодействие управляющих станций и тест-сервера в локальной сети осуществлялось через стандартные для операционной системы Windows 95 службы доступа к общим файлам и не требовало специальной программной поддержки.

Обмен данными между тест-сервером и рабочими станциями через Интернет по протоколу TCP выполняли программы ТСервер и Оператор, разработанные авторами работы в среде визуального программирования Delphi 3.0.

С помощью методики из банка тестовых заданий бланкового централизованного тестирования были скомпонованы и перенесены по локальной сети на тест-сервер в виде реляционной базы данных 10 компьютерных тестов по каждому из 28 предметов 9-го, 10-го и 11-го классов.

После завершения всех подготовительных работ Центр тестирования выпускников общеобразовательных учреждений РФ объявил о проведении бесплатного репетиционного тестирования через Интернет для учащихся учебных заведений и будущих абитуриентов.

С 3 октября 1998 г. по 31 марта 1999 г. с Web-сервера Центра тестирования (<http://www.vin.kaluga.ru/>) было выдано для установки на рабочих станциях 1184 копии программы Оператор и проведено 8255 сеансов компьютерного тестирования через Интернет, в ходе которых программа ТСервер, запущенная на тест-сервере, обработала 245 373 заявки на получение

тестового задания или на подсчёт набранных баллов. Больше всего заявок (12857, около 5% от общего количества) обработано 16 марта 1999 г. (на 165-е сутки эксперимента). Основная часть заявок (72% от общего количества) поступила на тест-сервер в марте, перед апрельским централизованным тестированием на бланках, причём по дням недели распределение заявок оказалось достаточно равномерным.

Из двух равных по длительности больший интерес представляет вторая часть эксперимента, относящаяся к периоду с 1 января 1999 г. по 31 марта 1999 г. На её пришлось 93,5% всех заявок.

Максимальная суточная нагрузка зарегистрирована 16 марта 1999 г — 12 857 заявок, из которых 1965 (около 15%) поступили в промежуток с 11:00 до 12:00 — абсолютный рекорд за всё время эксперимента. Характерным является суточный цикл — незначительное количество заявок в ночное время, рост активности в утренние часы, максимальная нагрузка в полдень, некоторый спад во время обеда, после которого снова всплеск активности с последующим постепенным снижением нагрузки вечером.

Аналогичные результаты получены ранее другими исследователями для интенсивности запросов к автоматизированной системе управления погрузочно-разгрузочными работами на морском транспорте в течение дня\* для интенсивности потока телефонных разговоров в течение суток\*\*.

\* Альянах И.Н. Моделирование вычислительных систем. Л.: Машиностроение, 1988.

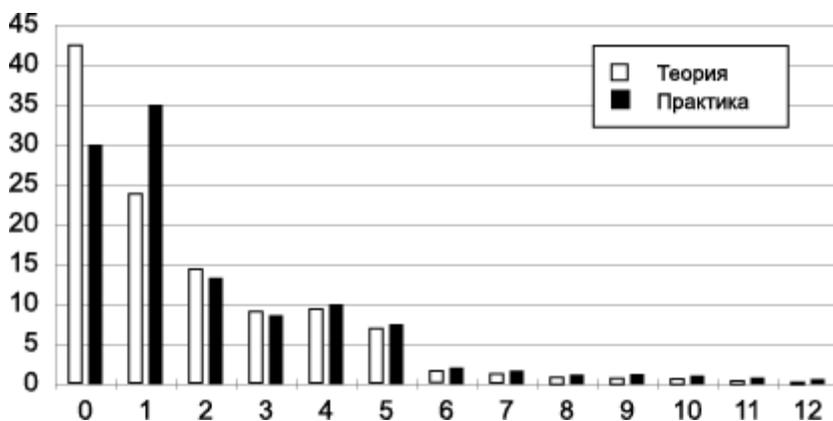
\*\* Майоров С.А., Новиков Г.И., Алиев Т.И., Махарёв Э.И., Тимченко Б.Д. Основы теории вычислительных систем: Учебное пособие для вузов. М.: Высшая школа, 1978.

Представленные для ИВС КТ распределения говорят о нестационарности потока заявок, поступающих на тест-сервер. Рабочая нагрузка на систему зависит от месяца года (быстро растёт по мере приближения к апрельскому централизованному тестированию на бланках), дня недели (увеличивается до среды, затем постепенно уменьшается), времени суток (достигает максимума в полдень).

В рамках данного исследования не ставилась задача проведения анализа временного ряда и подбора статистической модели для описания зависимости интенсивности поступления заявок от времени, так как пока собраны данные лишь за бмесяцев и на основе эксперимента, обладающего спецификой репетиционного тестирования выпускников общеобразовательных учреждений и будущих абитуриентов.

Для оценки работоспособности ИВС КТ в режиме пиковой нагрузки, которая в 34 раза превысила среднюю, рассмотрим распределение интервалов времени между моментами поступления на единственный тест-сервер заявок от рабочих станций.

**Распределение интервалов времени между заявками по длине (пиковая нагрузка — 16.03.1999 г. с 11:00 до 12:00)**



Для сравнения рядом с данными, полученными в ходе эксперимента (практика), показано экспоненциальное распределение (теория), характерное для интервалов времени между двумя последовательными заявками в случае простейшего потока\*, для которого в качестве интен-

сивности выбрано 10,55 заявки в секунду.

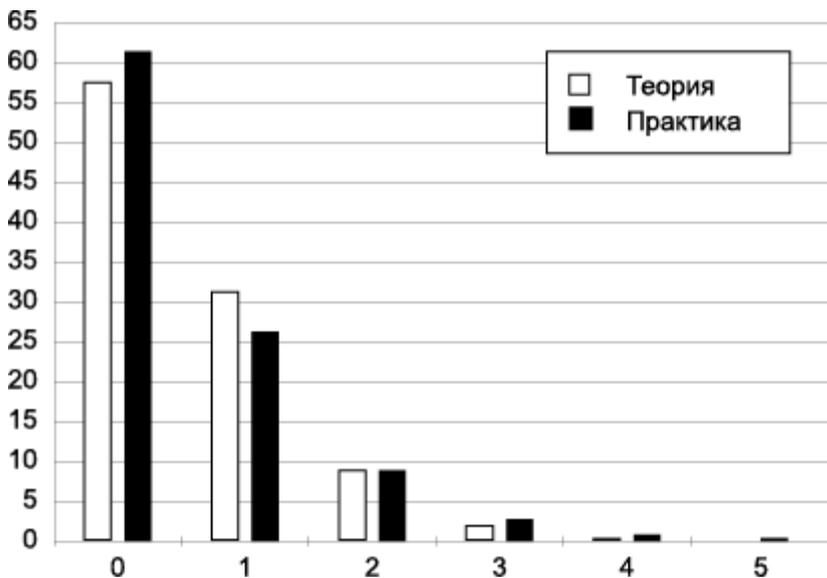
\* Майоров С.А., Новиков Г.И., Алиев Т.И., Махарёв Э.И., Тимченко Б.Д. Основы теории вычислительных систем: Учебное пособие для вузов. М.: Высшая школа, 1978.

Чтобы определить степень согласия между экспериментальным и экспоненциальным распределениями, вычислим статистику хи-квадрат Пирсона для простой гипотезы, приняв в качестве элементарных исходов попадание длины интервала времени между двумя последовательными заявками в один из 10 отрезков:  $[0; 1)$ ,  $[1; 2)$ ,  $[2; 3)$ ,  $[3; 4)$ ,  $[4; 5)$ ,  $[5; 6)$ ,  $[6; 7)$ ,  $[7; 8)$ ,  $[8; 9)$ ,  $[9; +\infty)$ . Получаем  $X = 67,8 + 78,0 + 0,6 + 0,6 + 0,4 + 1,6 + 0,7 + 0,2 + 0,0 + 138,4 = 288,4$  — слишком большое значение для распределения хи-квадрат с 9 степенями свободы, основной вклад в которое делают отрезки  $[0; 1)$ ,  $[1; 2)$  и  $[9; +\infty)$ .

В эксперименте зафиксировано преобладание интервалов времени между заявками с длиной от 1 до 2 секунд, реже встречались более короткие и имели место очень длинные интервалы (до 44 секунд), вероятность появления которых при экспоненциальном распределении мала (в простейшем потоке длина 99,9% интервалов времени между последовательными заявками меньше 13 секунд).

Ниже представлено экспериментальное распределение вероятности поступления  $K$  заявок за секунду (практика) в сравнении с распределением Пуассона при интенсивности стационарного потока  $\lambda = 0,55$  заявки в секунду (теория). Видно, что по результатам эксперимента вероятность не получить ни одной заявки в течение секунды оказывается больше, а получить одну заявку соответственно меньше, чем в случае пуассоновского потока.

**Распределение вероятности поступления  $K$  заявок за секунду (пиковая нагрузка — 16.03.1999 г. с 11:00 до 12:00)**



Таким образом зарегистрированный в эксперименте поток заявок, поступающих на тест-сервер, создаёт для ИВС КТ менее тяжёлый режим работы по сравнению с простейшим (стационарным пуассоновским) потоком. Поэтому используемое далее предположение о том, что на вход системы поступает простейший поток заявок, приведёт к определению предельных значений характеристик качества обслуживания (в нашем случае это среднее время отклика на запрос РС при ограничениях на производительность ТС и подсистемы передачи данных по каналам связи), и реальная система будет функционировать не хуже, чем это следует из полученных оценок\*.

\* Майоров С.А., Новиков Г.И., Алиев Т.И., Махарёв Э.И., Тимченко Б.Д. Основы теории вычислительных систем: Учебное пособие для вузов. М.: Высшая школа, 1978.

## Разработка и исследование аналитической модели блока обработки заявок, поступающих на тест-сервер ИВС КТ

В процессе компьютерного тестирования, в локальной сети или через Интернет, ТС получает от РС множество заявок и должен быстро их обрабатывать, располагая ограниченными ресурсами конкретного компьютера (быстродействие процессора, объём оперативной памяти, скорость обмена данными с жёстким диском и т.д.). Чтобы выбрать технические средства, адекватные поставленным задачам, и определить эффективность использования оборудования, необходимо оценить функциональные возможности тест-сервера ИВС КТ. Сделать это можно с помощью следующей аналитической модели блока обработки заявок, поступающих на тест-сервер от рабочих станций.

Введём обозначения:

$\lambda$  — интенсивность сеансов тестирования в режиме оперативной обработки;

$i = 1, \dots, M$  — типы поступающих оперативных заявок  $Z_i$ ;

$a_i$  — среднее число заявок типа  $i$ , поступающее в ходе одного сеанса;

$\lambda_i = \lambda a_i$  — интенсивность входного потока заявок типа  $i$ ;

$\lambda b$  — интенсивность сеансов тестирования в режиме отложенной обработки;

$Z_{M+1}$  — заявка на отложенную обработку пакета;

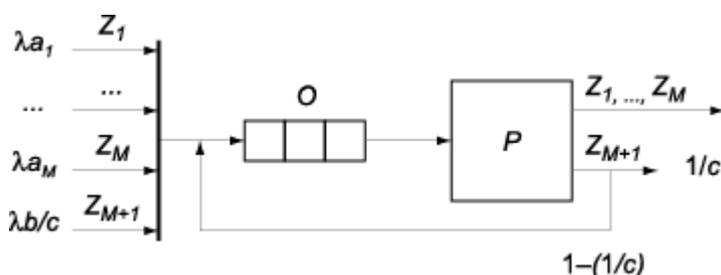
$c$  — среднее количество записей в одном пакете;

$\lambda_{M+1} = \lambda b/c$  — интенсивность входного потока заявок  $Z_{M+1}$ .

С учётом введённых обозначений работа блока обработки выглядит следующим образом.

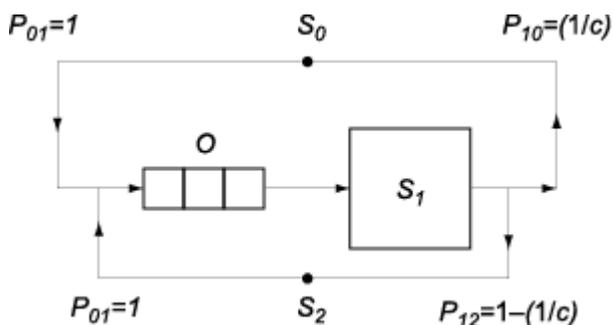
Все заявки поступают в неограниченную очередь  $O$  и ожидают обслуживания единственным процессором  $P$  сколько угодно долго. После обслуживания в порядке очереди (бесприоритетная дисциплина FIFO) заявки  $Z_i$  с вероятностью 1 покидают блок обработки, заявка  $Z_{M+1}$  на обработку пакета из  $c$  записей с вероятностью  $(1/c)$  покидает блок обработки, а с вероятностью  $1-(1/c)$  встаёт на дообслуживание в самый конец очереди  $O$ .

**Исходная модель неприоритетного обслуживания заявок в блоке обработки тест-сервера ИВС КТ**



Определим интенсивность суммарного потока пакетных заявок. Для этого представим блок обработки в виде стохастической сети, обозначив вероятность поступления заявки  $Z_{M+1}$  из системы  $S_j$  в систему  $S_i$  через  $p_{ji}$ .

**Стохастическая сеть блока обработки пакетных заявок  $Z_{M+1}$**



В стационарном режиме интенсивности входящего и выходящего потоков любой си-

стемы  $S_i$  одинаковы (без потерь обслуживаются все поступившие на обработку заявки) и равны сумме интенсивностей потоков, поступающих в неё из других систем. Это позволяет составить систему линейных уравнений\*:

\* Майоров С.А., Новиков Г.И., Алиев Т.И., Махарёв Э.И., Тимченко Б.Д. Основы теории вычислительных систем: Учебное пособие для вузов. М.: Высшая школа, 1978. 408 с.

$$\lambda_i = \sum_{j=0}^n \lambda_j p_{ji} \quad (i = 0, \dots, n)$$

В случае нашей стохастической сети система уравнений принимает вид:

$$\begin{cases} \lambda_0 = \lambda_0 0 + \lambda_1 (1/c) + \lambda_2 0; \\ \lambda_1 = \lambda_0 1 + \lambda_1 0 + \lambda_1 1; \\ \lambda_2 = \lambda_2 0 + \lambda_1 (1 - (1/c)) + \lambda_2 0; \end{cases}$$

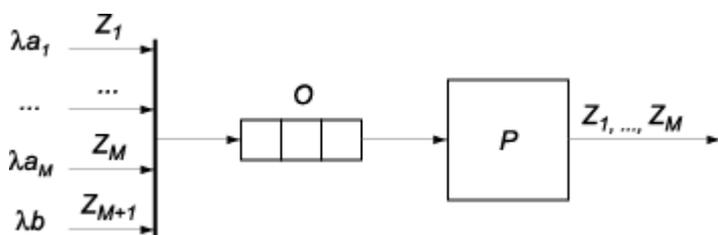
Из первого уравнения системы находим  $\lambda_1 = c\lambda_0$ . Для блока обработки это значит, что интенсивность суммарного потока заявок  $Z_{M+1}$  равна

$$\lambda_i \sum_{M+1} = c(\lambda_{M+1}) = c(\lambda b / c) = \lambda b$$

Результат примечателен тем, что констатирует отсутствие зависимости между средним количеством записей в одном пакете  $s$  и интенсивностью суммарного потока пакетных заявок  $Z_{M+1}$ .

Используя полученный результат, преобразуем исходную модель блока обработки тест-сервера.

#### Преобразованная модель беспriorитетного обслуживания заявок в блоке обработки тест-сервера ИВС КТ



Положив  $a_{M+1}$ , устраним последнее отличие пакетных заявок  $Z_{M+1}$ . Теперь каждый тип заявок формирует поток с интенсивностью  $\lambda_i = \lambda a_i$ , со средней длительностью обслуживания  $v_i$  и дисперсией  $\sigma_i^2$ , с коэффициентом загрузки блока обработки  $p_i = \lambda_i v_i$ .

Допустив, что каждый из потоков заявок одного типа является простейшим, получим одноканальную систему массового обслуживания  $M/G/1/$  с единственным процессором, неограниченной очередью и неограниченным временем ожидания\*. Условие существования в СМО стационарного режима

\* Альяных И.Н. Моделирование вычислительных систем. Л.: Машиностроение, 1988.

$$R = \sum_{j=1}^{M+1} p_j < 1$$

При его выполнении и использовании беспriorитетной дисциплины обслуживания FIFO среднее время ожидания в очереди для заявок всех типов одинаково и равно\*

\* Майоров С.А. и др. Указ. соч.

$$\omega = \frac{\sum_{j=1}^{M+1} \lambda(\nu_j^2 + \sigma_j^2)}{2(1-R)}$$

Оставшиеся основные характеристики функционирования блока обработки тест-сервера выражаются через уже полученные:

$u_i = \omega_i + \nu_i$  — среднее время пребывания заявки типа  $i$  в блоке обработки;

$\bar{u}_i = \frac{\sum_{j=1}^{M+1} \lambda_j u_j}{\sum_{j=1}^{M+1} \lambda_j}$  — среднее время пребывания одной заявки из суммарного обработки;

$\lambda_i = \lambda_i \omega_i$  — среднее число заявок типа  $i$  в очереди;

$l = \sum_{j=1}^{M+1} l_j$  — среднее число заявок

всех типов в очереди;

$n_i = \lambda_i u_i$  — среднее число заявок типа  $i$  в блоке обработки;

$\bar{n} = \sum_{j=1}^{M+1} n_j$  — среднее число заявок всех типов в блоке обработки.

Для оценки функциональных возможностей тест-сервера подставим в аналитическую модель параметры, полученные экспериментальным путём на персональном компьютере Pentium-166MHz/32MB с операционной системой Windows95 (табл.1). Заметим, что среднее время обслуживания заявок 3-го и 6-го типов значительно больше, так как их обработка требует добавления в базу данных новых записей и обновления индексов.

**Таблица 1. Параметры ИВС КТ, полученные экспериментальным путём**

- 1 — Заявка
- 2 — Тип  $i$
- 3 — Частота  $a_i$
- 4 — Среднее время обслуживания  $\vartheta_i$
- 5 — Загрузка процессора  $\rho_i$

1	2	3	4	5
Список предметов	1	1	0,003	0,003 $\lambda$
Список тестов	2	1	0,002	0,002 $\lambda$
Структура теста	3	1	0,027	0,027 $\lambda$
Задание	4	50	0,002	0,002 $\lambda$
Результат	5	1	0,003	0,003 $\lambda$
Пакет	6	13	0,030	0,030 $\lambda$

Суммарная загрузка единственного процессора со стороны всех потоков —  $R = 0,525\lambda$ . Из условия существования в СМО стационарного режима  $R < 1$  находим ограничение на интенсивность сеансов тестирования в режиме оперативной обработки  $\lambda \leq (1/0,525) \approx 1,9$  сеанса в секунду.

Изменение производительности компьютера, на котором работает тест-сервер, прямо пропорционально сказывается на критическом значении  $\lambda$ , при приближении к которому резко возрастает  $u_{\text{ср}}$ , среднее время пребывания заявки в блоке обработки. Видно, что уже при  $\lambda = 0,5$  сеанса в секунду (1800 сеансов в час, что невероятно много для on-line тестирования даже в масштабах страны) разница в значениях  $u_{\text{ср}}$  мала и не будет ощущаться участниками тестирования на компьютерах через Интернет.

Существенное влияние на загрузку процессора блока обработки тест-сервера  $\rho$  будет оказывать параметр  $b$  — соотношение между интенсивностью отложенной и оперативной обработки, определяя частоту появления пакетов из  $c$  записей (в среднем по  $\nu_6$  секунд на об-

служивание каждой записи пакета).

Параметр  $b$  принимает большие значения, когда основная часть сеансов тестирования проводится либо на компьютерах в изолированной локальной сети, либо на бумаге с последующим обращением к тест-серверу для проверки ответов и подсчёта набранных баллов сразу для нескольких тестируемых.

При использовании тест-сервера в централизованном и репетиционном тестировании через Интернет в 1999 году соотношение между числом сеансов отложенной и оперативной обработки было  $103\ 907:8255 \approx 13:1$  ( $a_6$  в табл.1), а максимальная (в среднем за час) интенсивность  $\lambda \approx 0,008$  сеанса в секунду.

Сравнивая оценки, полученные с помощью разработанной аналитической модели, и реальную нагрузку на блок обработки, можно сказать о наличии запаса производительности и об отсутствии необходимости наращивать вычислительную мощь тест-сервера, поскольку в данном случае он не является «узким местом» в отличие от телекоммуникационной подсистемы.

При других условиях предложенная аналитическая модель поможет оценить функциональные возможности тест-сервера ИВС КТ и выбрать технические средства, адекватные поставленным задачам.

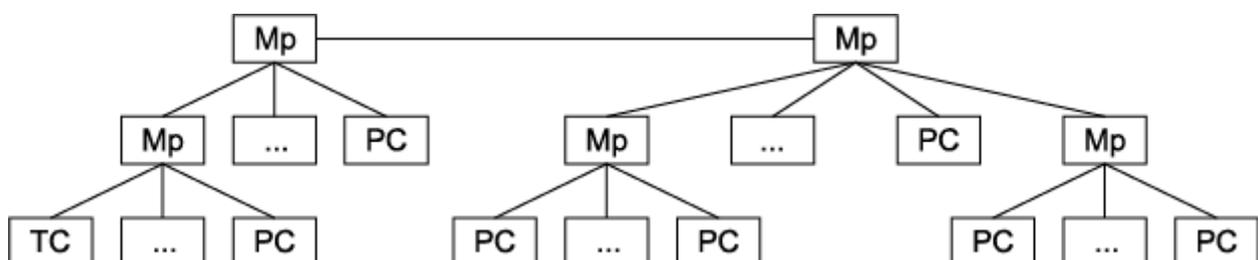
## Разработка и исследование имитационной модели телекоммуникационной подсистемы ИВС КТ

Концептуальная модель ИВС КТ предполагает использование эффективной подсистемы передачи данных по каналам связи, особенно при проведении сеансов тестирования через Интернет, когда единственным тест-сервером (ТС) одновременно обслуживается множество рабочих станций (РС) и передаются значительные объёмы информации. Взаимодействие рабочей станции и тест-сервера соответствует модели «клиент-сервер»\*, в которой клиент (РС) запрашивает информацию (тестовое задание) или услугу (проверку ответов и подсчёт набранных баллов), а сервер (ТС) отвечает на запрос.

\* Джамса К., Коуп К. Программирование для Internet в среде Windows. СПб.: Питер, 1996.

Поскольку все обращения к ТС со стороны РС инициирует пользователь (участник тестирования), который хочет получать ответы как можно быстрее, то в качестве показателя эффективности телекоммуникационной подсистемы ИВС КТ целесообразно принять среднее время отклика системы на запрос клиента. Определить его трудно, учитывая распределённость ИВС КТ и сложность схемы подключения рабочих станций к тест-серверу через маршрутизаторы (МР) по каналам глобальной сети Интернет.

**Схема подключения рабочих станций (РС) к тест-серверу (ТС) через маршрутизаторы (МР) по каналам глобальной сети Интернет**



Каждый канал связи (МР-ТС, МР-РС, МР-МР) характеризуется пропускной способностью (объёмом информации, который можно передать за единицу времени), задержкой (временем, которое уходит на работу аппаратуры и распространение сигналов в среде передачи данных) и загруженностью (процентом использования пропускной способности канала и очереди пакетов в маршрутизаторе несколькими одновременно работающими пользовате-

лями).

Производительность виртуальной цепи передачи данных между РС и ТС, формируемой всеми промежуточными каналами связи и маршрутизаторами, во многом определяется «узким местом» цепи: перегруженным каналом общего пользования, маршрутизатором с неадекватным буфером для очереди пакетов, низкоскоростной модемной линией для подключения конечного пользователя к аппаратуре поставщика Интернет-услуг и так далее.

Получить информацию о виртуальной цепи между двумя узлами сети можно с помощью утилит: Ping, TracerT, ICB\*. Программа Ping использует стандартную функцию Echo протокола ICMP\*\* для посылки небольшого сообщения удалённому компьютеру или маршрутизатору и получения от него идентичного ответа.

---

\* Carter R.L., Crovella M.E. Measuring Bottleneck Link Speed in Packet-Switched Networks. Technical Report B-CS-96-006. Boston, MA: Computer Science Department, Boston University, 1996.

\*\* Postel J. Internet Control Message Protocol. RFC 793. Information Sciences Institute, University of Southern California, 1981.

Она может:

- вставляя индивидуальные номера и контрольные суммы в IP-пакеты, зафиксировать их потерю, повторение, перестановку или искажение;
- указывая в IP-пакете время отправления сообщения и сравнивая его с временем получения ответа, вычислить продолжительность путешествия в обе стороны (RTT — Round Trip Time).

Программа Ping не может:

- объяснить причины потери, повторения, перестановки или искажения IP-пакетов и определить точку сбоя в сети;
- показать временную диаграмму прохождения IP-пакета по каналам связи через промежуточные маршрутизаторы.

В качестве примера показаны результаты работы программы Ping по виртуальной цепи между узлами [www.vin.kaluga.ru](http://www.vin.kaluga.ru) и [www.info-art.ru](http://www.info-art.ru).

Pinging [www.infoart.ru](http://www.infoart.ru) [195.161.0.153] with 32 bytes of data:

```
Reply from 195.161.0.153: bytes=32 time=99ms TTL=246
Reply from 195.161.0.153: bytes=32 time=68ms TTL=246
Reply from 195.161.0.153: bytes=32 time=63ms TTL=246
Reply from 195.161.0.153: bytes=32 time=47ms TTL=246
Reply from 195.161.0.153: bytes=32 time=84ms TTL=246
Reply from 195.161.0.153: bytes=32 time=45ms TTL=246
Reply from 195.161.0.153: bytes=32 time=60ms TTL=246
Reply from 195.161.0.153: bytes=32 time=33ms TTL=246
Reply from 195.161.0.153: bytes=32 time=41ms TTL=246
Reply from 195.161.0.153: bytes=32 time=42ms TTL=246
```

Ping statistics for 195.161.0.153:

```
Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 33ms, Maximum = 99ms, Average = 58ms.
```

Видно, что узлу [www.infoart.ru](http://www.infoart.ru) с IP-адресом 195.161.0.153 послано 10 пакетов по 60 байт (20 байт под IP-заголовок + 8 байт под ICMP-сообщение + 32 байта под данные). На каждую посылку через промежуток времени *time* получен ответ. Потерь нет. Минимальное время прохождения одного пакета в обе стороны (RTT) равно 33 мс, максимальное — 99 мс, а среднее — 58 мс.

Значение RTT изменяется относительно некоторой константы, связанной с работой аппаратуры и распространением сигналов в среде передачи данных, в пределах диапазона, ширина которого зависит от загруженности каналов и маршрутизаторов, демонстрируя стохастичность виртуальной цепи.

Перечень промежуточных маршрутизаторов между двумя узлами сети можно определить с помощью программы TracerT, которая посылает в адрес удалённого компьютера IP-пакеты, указывая в поле TTL (Time To Live) порядковые номера. При прохождении IP-пакета через каждый промежуточный маршрутизатор значение поля TTL уменьшается на единицу. При достижении нуля в поле TTL продвижение IP-пакета прекращается, а его отправителю посылается ICMP-сообщение Time Exceeded с указанием IP-адреса текущего маршрутизатора. Таким образом, программа TracerT по порядку получает IP-адреса всех промежуточных маршрутизаторов, перечень которых между узлами www.vin.kaluga.ru и www.infoait.ru представлен в качестве примера.

Tracing route to www.infoart.ru [195.161.0.153]:

```

##      ms      ms      ms
 1      1       1       2   137-5.town-net.obninsk.com [195.90.137.5]
 2      7       7       6   gw3.town-net.obninsk.com [195.90.137.4]
 3     12      10      10  stranger-r.obninsk.com [195.90.137.3]
 4     11      12      13  layka-r.obninsk.com [195.90.137.2]
 5     19      19      53  herkules-r.kaluga.ru [195.90.136.3]
 6     64      38      48  Kaluga-gw.Capella.Rosnet.Net [195.90.159.13]
 7    134      31      37  195.90.129.22
 8     72     108     89  MSK-M9-IX.rt.ru [195.232.244.150]
 9    129     105     91  www.infoart.ru [195.161.0.153]

```

Каждый IP-пакет через маршрутизатор Viper-DSL фирмы 3Com\* сначала попадает в компьютерную сеть города Обнинска, потом через Калугу поступает на маршрутизатор компании Роснет, затем следует до точки М9 и заканчивает свой путь в пункте назначения. Этот маршрут достаточно стабилен и меняется только в аварийных ситуациях.

\* VIPER-DSL and Affinity LC. Release 1 Installation Guide. Skokie, IL: 3civ Corp., 1997. 53 p.

С помощью утилит Ping и TracerT можно оценить загруженность и удалённость отдельных участков виртуальной цепи между двумя узлами, однако для определения её пропускной способности лучше воспользоваться программой ICB, первоначально разработанной для контроля за пропускной способностью магистральных каналов одного из крупнейших в Голландии поставщиков Интернет-услуг [<http://www.byzantine.nl/icb/>].

ICB использует протокол ICMP и учитывает следующие факторы:

- время необходимое для подключения к удалённому узлу;
- время необходимое для посылки пакета с данными;
- фрагментацию и размер полученного пакета;
- время необходимое для возврата пакета с данными;
- фрагментацию и размер возвращённого пакета.

Программа динамически изменяет размер пакета данных и периодически повторяет измерения, по результатам которых отображает среднее значение пропускной способности виртуальной цепи (табл. 2.).

**Таблица 2. Результаты работы программы ICB**

Город	Сервер	RTT, мс	Средняя пропускная способность, Кбит/с
Обнинск	www.obninsk.com	9	820
Калуга	www.kaluga.ru	16	484
Москва	www.msu.ru	34	328
Санкт-Петербург	www.spb.ru	71	193
Уфа	www.ufatu.ac.ru	72	211
Красноярск	www.krasnoyarsk.ru	81	237
Нью-Йорк	www.mci.com	252	89

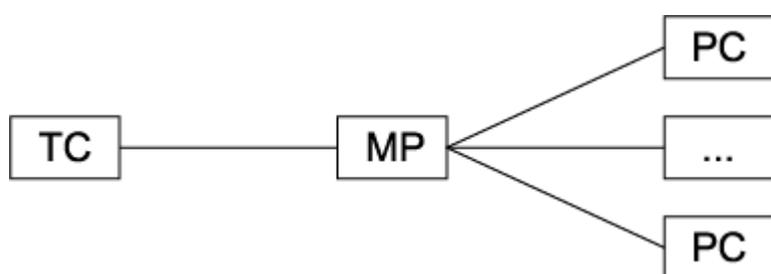
На основе исследования различных виртуальных цепей передачи данных между РС и ТС, проведённого с помощью утилит Ping, TracerT, ICB, можно сделать вывод, что большинство маршрутов следования информации проходит по двум лучам распределённой звезды с цен-

тром в точке М9, каждый из которых, как правило, включает каналы одной из крупных московских компаний, регионального провайдера, местного поставщика Интернет-услуг и конечного пользователя.

У любого луча есть «узкие места», которые определяют его пропускную способность и суммарную задержку. В случае тест-сервера таковыми являются самый медленный (номинально или вследствие перегрузки) канал связи и выходящий на него со стороны тест-сервера (в этом направлении передаётся значительно больше информации) маршрутизатор. Для луча, ведущего от точки М9 к рабочей станции, «узкими местами» бывают либо перегруженный канал связи общего пользования, либо низкоскоростная модемная линия, часто используемая для подключения к аппаратуре местного поставщика Интернет-услуг.

Такое положение дел позволяет упростить схему телекоммуникационной подсистемы ИВС КТ, заменив лучи едиными каналами с соответствующими характеристиками, которые объединяются одним маршрутизатором, стоящим сразу за самым медленным участком виртуальной цепи ТС-М9.

#### Упрощённая схема телекоммуникационной подсистемы ИВС КТ



Чтобы оценить работоспособность предложенной выше упрощённой схемы телекоммуникационной подсистемы ИВС КТ, необходимо знать зависимости характеристик (времени отклика на запрос клиента, процента использования пропускной способности канала сервера, длины очереди в маршрутизаторе) от параметров (пропускных способностей и задержек каналов, объёма передаваемой информации, количества одновременно работающих клиентов).

С помощью аналитического моделирования можно представить маршрутизаторы в виде систем массового обслуживания. Такой подход не позволит учесть такие важные особенности реальной сети, как ограниченность очередей, потери пакетов, паузы, повторные передачи, механизмы восстановления после ошибок и алгоритмы управления перегрузкой.

При натуральном эксперименте можно либо анализировать информацию, собранную в регистрационных журналах за некоторый период реальной эксплуатации системы, либо пытаться искусственно формировать обращения клиентов к серверу, исходя из представлений о характере реальной нагрузки. В первом случае необходимо обосновать репрезентативность выбранного периода, во втором — адекватность используемых представлений. В любом случае трудно будет предсказывать последствия изменения хотя бы одного из параметров.

При имитационном моделировании можно использовать алгоритмы, наиболее полно описывающие работу технических и программных средств. Разработка и проверка подобных алгоритмов требуют специальных знаний, практического опыта и значительных трудозатрат. Сам процесс имитационного моделирования предполагает предварительное планирование и многочисленные прогоны при разных значениях параметров с последующей статистической обработкой результатов.

Для данного исследования был выбран последний метод во многом из-за авторитетного средства имитационного моделирования сетей — пакета Network Simulator v.2.0, созданного в ноябре 1998 года в рамках совместного проекта UC Berkeley, LBL, USC/ISI и Xerox PARC\*.

\* Fall K., Varadhan K. NS Notes and Documentation. The VINT Project, 1998.

Пакет свободно распространяется в виде исходных текстов на языке C++, после компиляции которых получается небольшой исполняемый модуль-интерпретатор объект-

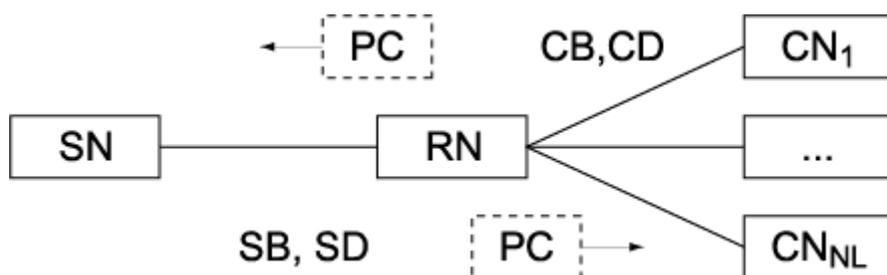
но-ориентированных сетевых имитационных моделей, написанных на языке TCL. Имеется документация и множество примеров.

Network Simulator предоставляет:

- готовые к использованию модели узлов, каналов, маршрутизаторов, сетевых протоколов и прикладных программ;
- вспомогательные элементы, среди которых агенты, очереди, таймеры, генераторы случайных чисел, трассировщики и мониторы;
- эмулятор работы в реальной сети и средство визуализации (в пакете Network Simulator).

При разработке имитационной модели телекоммуникационной подсистемы ИВС КТ, описание которой было выполнено на языке TCL, для создания сетевой топологии, соответствующей упрощённой схеме, использовались;

#### Сетевая топология телекоммуникационной подсистемы ИВС КТ



- узел маршрутизатора RN (Router Node);
- узел для тест-сервера SN (Server Node);
- дуплексный канал связи SN-RN с пропускной способностью SB (Server Bandwidth) и задержкой SD (Server Delay);
- узлы рабочих станций CN<sub>n</sub> (Client Node), где n = 1..NL (Node Limit);
- дуплексные каналы связи CN<sub>n</sub>-RN с пропускной способностью CB (Client Bandwidth) и задержкой CD (Client Delay).

Для передачи запросов клиента CR (Client Request) и ответов сервера SR (Server Response) по протоколу TCP каждой паре CN<sub>n</sub>-SN назначались агенты CT<sub>n</sub> (Client TCP) и ST<sub>n</sub> (Server TCP) типа FullTCP, управляемые приложениями CAn (Client Application) и SAn (Server Application) типа TCPApp.

Промежутки времени между получением от сервера очередного ответа и посылкой ему следующего запроса определялись независимо для каждого клиента с помощью случайных величин CW<sub>c</sub> (Client Wait), где c— 1..CL (Client Limit), имеющих экспоненциальное распределение и среднее значение AW (Average Wait)\*.

\* Тюрин Ю.Н., Макаров А.А. Статистический анализ данных на компьютере. М.: ИНФРА-М, 1998.

Продолжительность моделирования FT (Finish Time) устанавливалась так, чтобы при любом соотношении основных параметров имитационной модели телекоммуникационной подсистемы ИВС КТ (табл. 3) успевало пройти достаточное количество (несколько сотен) сеансов обмена данными между клиентами (рабочими станциями) и сервером (тест-сервером).

**Таблица 3. Основные параметры имитационной модели/телекоммуникационной подсистемы ИВС КТ**

Параметр	Комментарий	Значение
SB	Пропускная способность канала сервера	256 Кбит/с
SD	Задержка канала сервера	50 мс
SR	Размер ответа сервера	2*1024 байт
CB	Пропускная способность канала клиента	32 Кбит/с
CD	Задержка канала клиента	100 мс
CR	Размер запроса клиента	16 байт
CL	Количество клиентов	100 чел.
AW	Среднее время выполнения задания	2 мин.

В пакете Network Simulator моделируемая ИВС КТ рассматривалась как стохастическая система с дискретным изменением состояний, для которой случайные величины формировались за счёт преобразования выходного потока LCG( $2^{31}-1$ , 16 807, 0,1), носящего название MINSTD, а синхронизация параллельно протекающих во времени процессов выполнялась с помощью глобального списка особых событий.

Особые события, сигнализирующие о возможности перехода к следующей операции или о необходимости пересчёта характеристик модели, фиксировались трассировщиками и callback-функциями. Например, сразу после получения запроса клиента (завершения работы функции `mysend`) отправлялся ответ сервера (вызывалась функция `myrpost`), а все изменения в очереди пакетов на канале связи SN-RN со стороны тест-сервера регистрировались функцией `mydump`, которая рассчитывала загруженность очереди и канала.

Первоначально имитационную модель телекоммуникационной подсистемы ИВС КТ запускали в отладочном режиме с небольшим числом узлов и клиентов (например, при  $NL=2$  и  $CL=2$ ), чтобы проследить за динамикой процесса с помощью средства визуализации, которое позволяло:

- видеть узлы, каналы и их характеристики — пропускную способность, задержку и загруженность в зависимости от модельного времени;
- выбирать подходящие масштабы времени и изображения;
- контролировать и устанавливать текущее модельное время;
- приостанавливать, продолжать и повторять имитацию;
- наблюдать за продвижением всех пакетов, зная их последовательные номера, время отправки и размеры;
- отслеживать состояние очередей в маршрутизаторах и фиксировать моменты потери отдельных пакетов.

В дополнение к визуальному контролю выполнялась трассировка каналов, в ходе которой фиксировались события, связанные с прохождением IP-пакетов по виртуальной цепи между отправителем и получателем через промежуточные маршрутизаторы. Ниже показан пример трассировки дуплексного канала между тест-сервером (SN — узел 0) и маршрутизатором (RN — узел1) в момент обслуживания запроса клиента (CN<sub>1</sub>- узел 2) на получение тестового задания размером 2Кбайта.

#### Пример трассировки дуплексного канала SN-RN

**A** — факт поступления пакета в очередь входного («+») или выходного («г») маршрутизатора;

**B** — модельное время в секундах;

**C** и **D** — входной и выходной маршрутизаторы;

**E** — тип пакета;

**F** — его размер в байтах;

**G** — IP-поток;

**H** — отправитель;

**I** — получатель;

**J** — контрольный номер в пакете;

**K** — номер пакета.

A	B	C	D	E	F	G	H	I	J	K
+	33,7383	1	0	tcp	40	1	2,0	0,0	0	0
г	33,7896	1	0	tcp	40	1	2,0	0,0	0	0
+	33,7896	0	1	ack	40	1	0,0	2,0	0	1
г	33,8408	0	1	ack	40	1	0,0	2,0	0	1
+	34,0608	1	0	tcp	40	1	2,0	0,0	1	2
+	34,0748	1	0	tcp	56	1	2,0	0,0	1	3
г	34,1121	1	0	tcp	40	1	2,0	0,0	1	2
г	34,1266	1	0	tcp	56	1	2,0	0,0	1	3
+	34,1266	0	1	ack	552	1	0,0	2,0	1	4
г	34,1938	0	1	ack	552	1	0,0	2,0	1	4
+	34,5418	1	0	tcp	40	1	2,0	0,0	17	5
г	34,5931	1	0	tcp	40	1	2,0	0,0	17	5
+	34,5931	0	1	ack	552	1	0,0	2,0	513	6

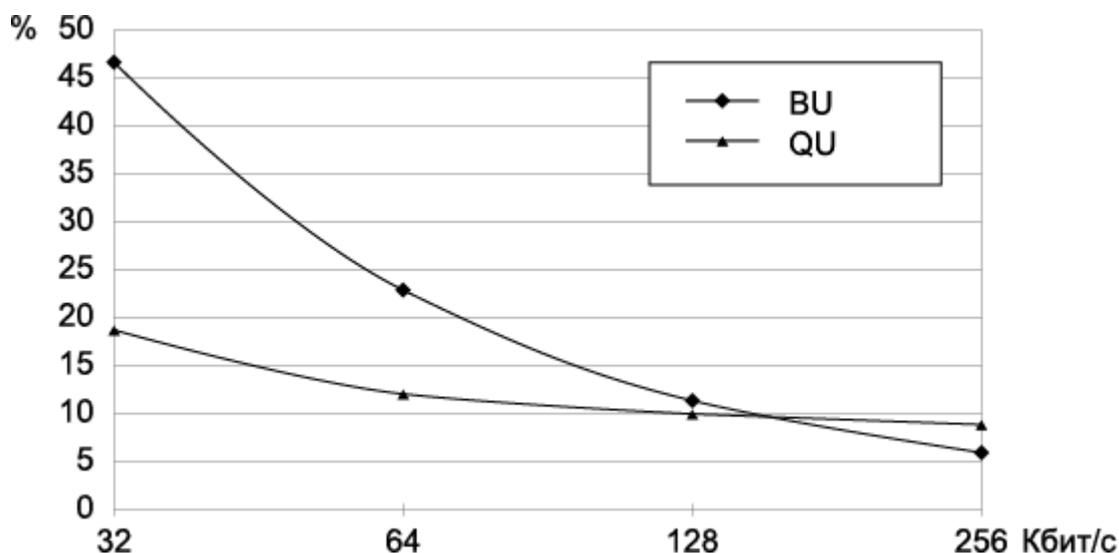
+	34,5931	0	1	ack	552	1	0,0	2,0	1025	7
r	34,6603	0	1	ack	552	1	0,0	2,0	513	6
r	34,6776	0	1	ack	552	1	0,0	2,0	1025	7
+	35,0083	1	0	tcp	40	1	2,0	0,0	17	8
r	35,0596	1	0	tcp	40	1	2,0	0,0	17	8
+	35,0596	0	1	ack	552	1	0,0	2,0	1537	10
r	35,1268	0	1	ack	552	1	0,0	2,0	1537	10
+	35,1463	1	0	tcp	40	1	2,0	0,0	17	9
r	35,1976	1	0	tcp	40	1	2,0	0,0	17	9
+	35,4748	1	0	tcp	40	1	2,0	0,0	17	11
r	35,5261	1	0	tcp	40	1	2,0	0,0	17	11
+	35,5261	0	1	ack	40	1	0,0	2,0	2049	12
+	35,5261	0	1	ack	40	1	0,0	2,0	2049	13
r	35,5773	0	1	ack	40	1	0,0	2,0	2049	12
r	35,5786	0	1	ack	40	1	0,0	2,0	2049	13
+	35,8073	1	0	tcp	40	1	2,0	0,0	18	14
r	35,8586	1	0	tcp	40	1	2,0	0,0	18	14

Визуальный контроль за динамикой процесса и анализ трассировки каналов показали, что имитационная модель при небольшом числе узлов и клиентов работает в соответствии с известными алгоритмами протокола TCP и выдает результаты, близкие к данным, полученным на основе исследования реальных каналов с помощью утилит NetStat и TcpDump. Таким образом, была подтверждена правильность программной реализации разработанной в рамках пакета Network Simulator имитационной модели телекоммуникационной подсистемы ИВС КТ.

Имитационная модель телекоммуникационной подсистемы ИВС КТ имеет восемь основных параметров, для каждого из которых в табл.3 представлено типовое значение, выбранное в результате анализа проведённых экспериментов и имеющегося опыта. Обсуждая результаты имитационного моделирования, будем полагать, что не указанные на оси абсцисс или в легенде параметры принимали типовые значения.

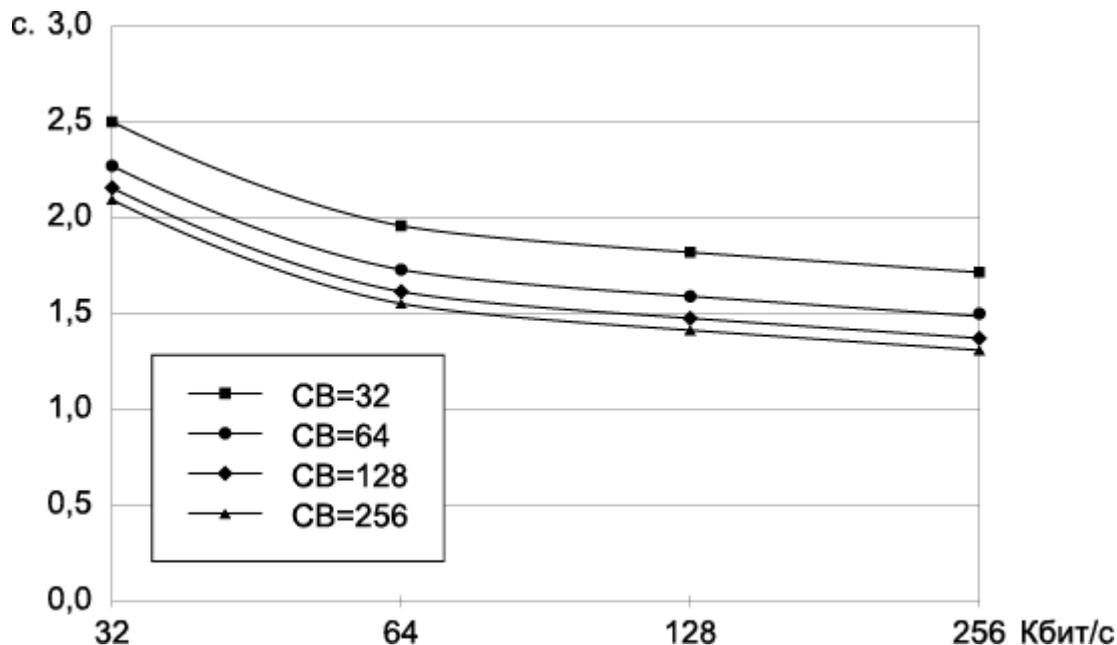
Сначала рассмотрим зависимость процента использования пропускной способности канала SN-RN (BU — Bandwidth Usage) и очереди пакетов в узле SN (QU — Queue Usage) от пропускной способности канала сервера SB. Видно, что при SB = 256 Кбит/с канал SN-RN и маршрутизатор SN загружены слабо — BU=6%, QU = 8%. Снижение пропускной способности канала приводит к увеличению загруженности — при SB=32 Кбит/с получаем BU=46%, QU=18%.

**Зависимость процента использования пропускной способности канала SN-RN (BU — Bandwidth Usage) и очереди пакетов в узле SN (QU — Queue Usage) от пропускной способности канала сервера SB**



Чем больше загруженность канала, тем эффективнее он используется с точки зрения экономики. С другой стороны, перегрузка канала негативно сказывается на значении показателя эффективности ИВС КТ, в качестве которого выступает среднее время отклика системы на запрос клиента RT (Response Time).

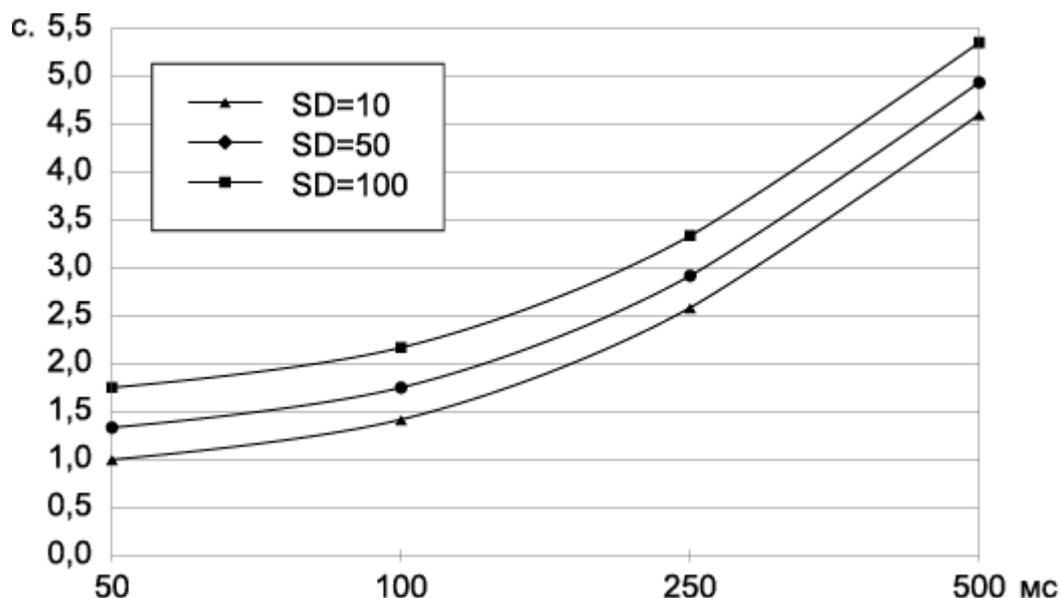
**Зависимость среднего времени отклика на запрос клиента RT от пропускной способности канала сервера SB при разной пропускной способности канала клиента CB (Кбит/с)**



Смена самой медленной (SB=32 Кбит/с, CB=32 Кбит/с) конфигурации на самую быструю (SB=256 Кбит/с, CB=256 Кбит/с) уменьшает среднее время отклика системы на запрос клиента почти в два раза — с 2,5 секунды до 1,3.

Не менее важным фактором, влияющим на показатель эффективности, являются задержки каналов, которые добавляются к времени доставки каждого IP-пакета вне зависимости от его размера и функций.

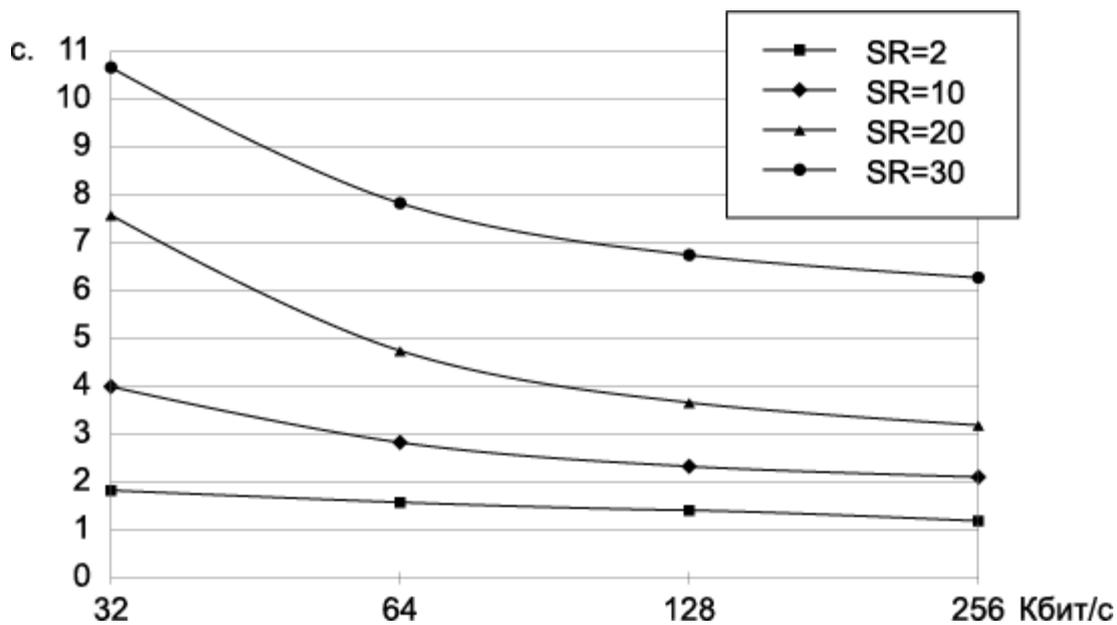
**Зависимость среднего времени отклика на запрос клиента RT от задержки канала клиента CD при разной задержке канала сервера SD (мс)**



Видно, что увеличение задержки канала клиента с 250 до 500 мс приводит к возрастанию времени отклика RT на 60% — с 3,3 секунды до 5,3.

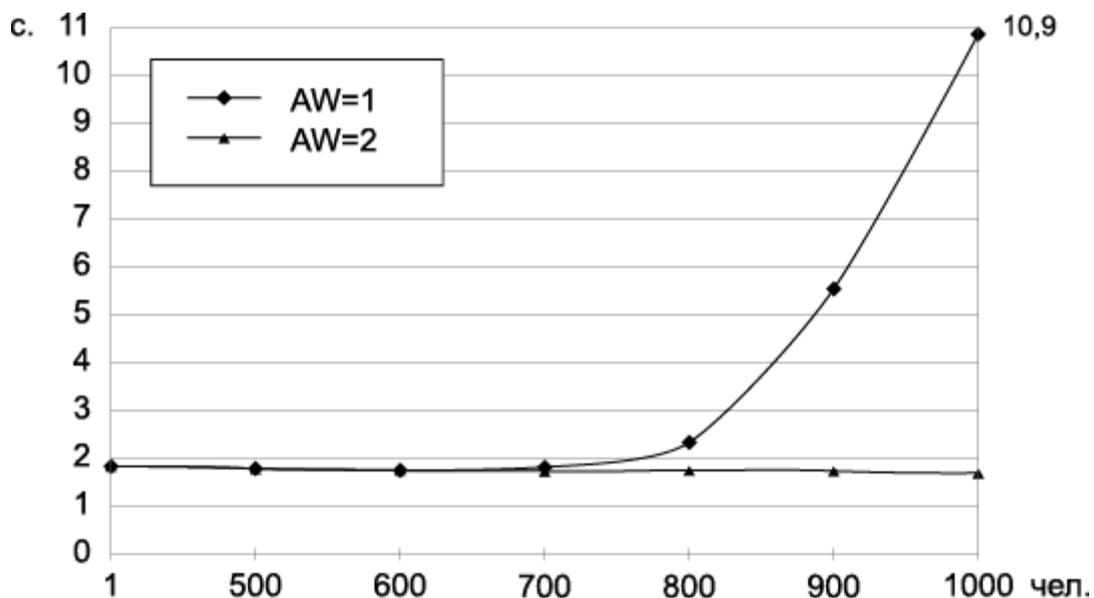
**Зависимость среднего времени отклика на запрос клиента RT от пропускной способности**

канала клиента СВ при разном размере ответа сервера СВ при разном размере ответа сервера SR (Кбайт)



В случае ИВС КТ большинство ответов сервера содержат запрошенные клиентом задания теста, поэтому параметр SR, принимающий значения от пары килобайт до нескольких мегабайт, можно рассматривать как средний размер тестового задания. В ситуации, когда  $SR > 10$  Кбайт, даже каналы  $SB = 256$  Кбит/с и  $CB = 256$  Кбит/с не смогут обеспечить время отклика  $RT < 2$  секунд для 100 клиентов, работающих одновременно. При  $SR = 2$  Кбайта телекоммуникационная подсистема ИВС КТ сможет обслуживать одновременно до 750 клиентов с приемлемым временем отклика  $RT < 2$  секунд, даже если среднее время выполнения одного задания AW сократится с 2 до 1 минуты.

**Зависимость среднего времени отклика на запрос клиента RT от количества работающих клиентов CL при разном среднем времени выполнения задания AW (мин)**



Таким образом, разработанная имитационная модель позволяет оценить функциональные возможности телекоммуникационной подсистемы ИВС КТ и выбрать соответствующие технические средства, чтобы обеспечить передачу данных между рабочими станциями и тест-сервером при взаимодействии, центральные алгоритмы которого рассматриваются ниже.

## Центральные алгоритмы взаимодействия рабочих станций и тест-сервера в рамках ИВС КТ

Взаимодействие рабочей станции и тест-сервера соответствует модели «клиент-сервер», в которой клиент (PC) запрашивает информацию (тестовое задание) или услугу (проверку ответов и подсчёт набранных баллов), а сервер (ТС) отвечает на запрос. Ниже в качестве примера показана временная диаграмма процесса получения информации объёмом 2 Кбайта клиентом, работающим через модем с пропускной способностью 3.6 Кбит/с и задержкой 100 мс, от удалённого сервера, использующего аппаратуру ADSL с пропускной способностью 256 Кбит/с и задержкой 50 мс.

**Диаграмма процесса получения информации объёмом 2 Кбайта через модем с пропускной способностью 33,6 Кбит/с и задержкой 100 мс**

Клиент	Время, мс	Сервер
	0	
1. Подготовка	1	
	1	
2. Открытие канала	149	...
3. Запись запроса	0	150
	...	
	250	
	3	1. Открытие канала
4. Ожидание ответа	1024	253 0
	1	2. Чтение запроса
	254	3. Регистрация запроса
	0	4. Обработка запроса
	...	
	1174	
5. Чтение ответа	667	...
6. Закрытие канала	0	1686
7. Сохранение ответа	2	5. Запись ответа
	1842	
	...	
	1940	
	3	6. Закрытие канала
	1943	
	1	7. Регистрация ответа
	1944	

Видно, что инициатором является клиент, который готовит исходные данные, открывает канал, записывает запрос и переходит в состояние ожидания ответа сервера. Получив соответствующий сигнал, сервер открывает канал со своей стороны, читает запрос клиента, регистрирует и обрабатывает его должным образом, записывает сформированный ответ и ждёт завершения сеанса связи.

Дождавшись ответа сервера, клиент читает его, закрывает канал и продолжает свою работу, предварительно сохранив ответ. Сервер получает извещение о завершении сеанса связи, закрывает канал со своей стороны, регистрирует выданный ответ и возвращается к обслуживанию других запросов, если они есть.

**Алгоритм функционирования рабочей станции в режиме оперативной обработки ответов тестируемых**



Он обеспечивает проведение тестирования за компьютером через Интернет непосредственно на тест-сервере и позволяет за счёт уменьшения объёма и продолжительности передачи данных:

- сократить финансовые затраты клиента, если он платит за сетевой трафик или за время подключения к сети;
- снизить нагрузку на сеть и на единственный тест-сервер;
- увеличить число клиентов, обслуживаемых одновременно с приемлемым временем отклика системы.

После запуска программы и ввода пароля предоставляется возможность либо пройти процедуру регистрации, выбрать новый тест и получить на тест-сервере идентификатор нового сеанса, либо восстановить сохранённый сеанс и продолжить работу над прерванным тестом.

Выполнение теста заключается в выборе правильного ответа на каждое тестовое задание, полученное на тест-сервере. Все тестовые задания передаются в сжатом виде. Распаковка выполняется в оперативной памяти рабочей станции непосредственно перед выводом на экран. Загруженное один раз через сеть задание сохраняется на диске так, что при следующем обращении к нему нет необходимости в его повторной загрузке с тест-сервера.

Для уменьшения сетевого графика алгоритм позволяет работать только с интересующими тестовыми заданиями, отобранными через структуру теста, не загружая с сервера ничего лишнего. Увидев на рабочем экране нужные задания (загрузив их тем самым в дисковый буфер), клиент может отключиться от сети (если оплачивается время подключения к Интернет) и спокойно выполнять отобранные задания столько времени, сколько это необходимо.

Следующее подключение к Интернет потребуется только для проверки ответов и подсчёта набранных баллов или для подгрузки заданий другого раздела теста.

В ситуациях, когда необходимо выполнить тест целиком, алгоритм даёт возможность подгрузить недостающие тестовые задания в дисковый буфер рабочей станции и продолжить тестирование в рамках компьютера клиента без обращений к серверу через Интернет.

После выполнения заданий теста предполагается проверка ответов на тест-сервере, который по запросу рабочей станции подсчитывает количество и составляет список правильно решённых заданий, определяет число набранных баллов и сообщает их клиенту, позволяя:

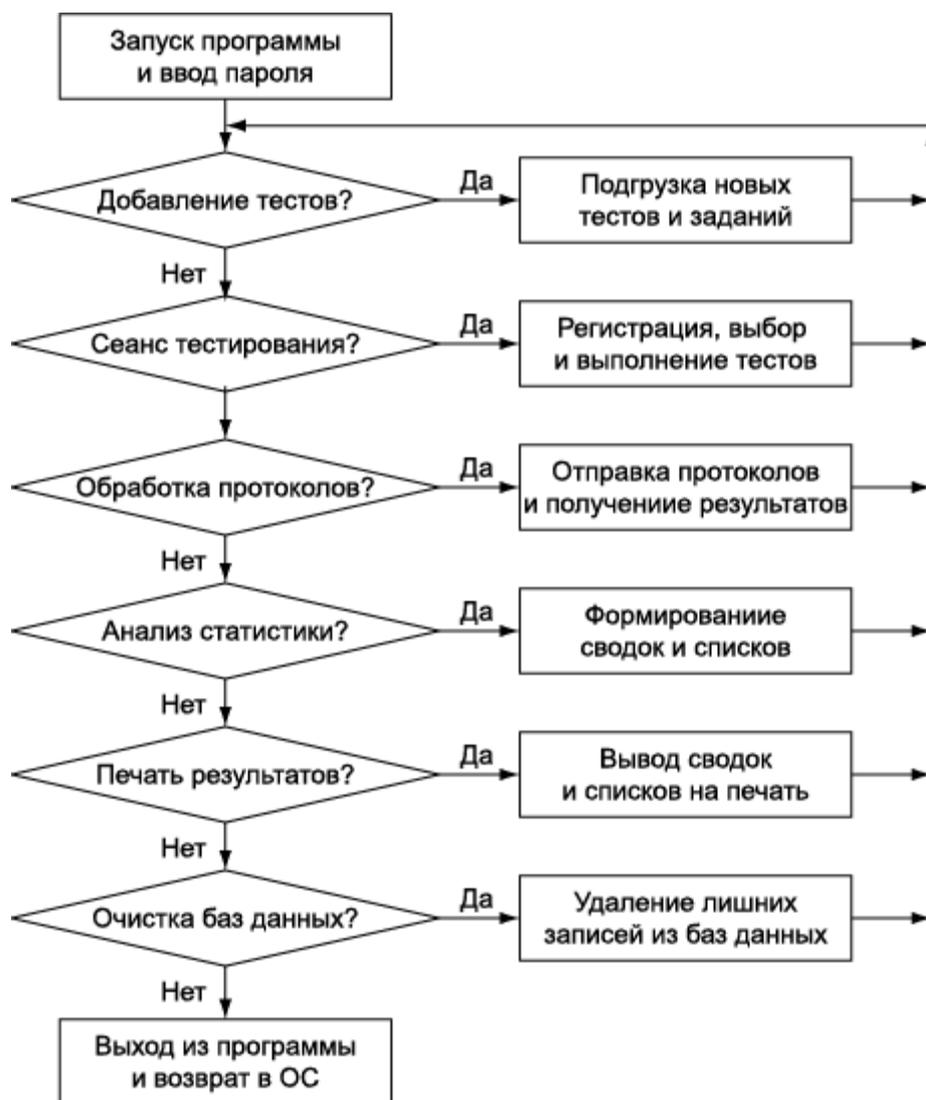
- видеть по количеству правильно и неправильно выполненных заданий степень освоения отдельной темы;
- группировать задания, относящиеся к интересующему разделу предмета, для тщательной его проработки;
- вернуться к неправильно выполненным заданиям теста для выявления и исправления допущенных ошибок или приступить к пропущенным ранее тестовым заданиям.

В учебных целях (например, в случаях репетиционного тестирования или самоконтроля) допускается продолжение работы с тестом после получения предварительных результатов. При проведении аттестационного тестирования такая возможность в алгоритме блокируется.

Выполнение теста может быть приостановлено по желанию тестируемого (с разрешения инструктора) или по техническим причинам (сбой в аппаратуре или в программном обеспечении компьютера клиента). Алгоритм обеспечивает продолжение работы с тестом с того места, где она была прервана, за счёт сохранения полной информации о сеансе, включая ответы и счётчики времени.

Завершить работу с тестом можно в любой момент (с разрешения инструктора). В отличие от приостановки в этом случае нельзя продолжить выполнение теста, так как вся информация о сеансе удаляется с рабочей станции сразу после получения соответствующего подтверждения от клиента.

**Алгоритм функционирования рабочей станции в режиме отложенной обработки ответов тестируемых**



Он обеспечивает проведение тестирования на компьютерах, объединённых локальной сетью, и даёт возможность за счёт совместного использования единственного комплекта программных модулей и баз данных:

- снизить расход дискового пространства;
- уменьшить трудозатраты на администрирование комплекса;
- облегчить контроль за работой тестируемых со стороны инструктора;
- упростить формирование протоколов тестирования для отправки через Интернет на обработку центральному тест-серверу;
- ускорить получение и анализ статистической информации о результатах проведённого тестирования.

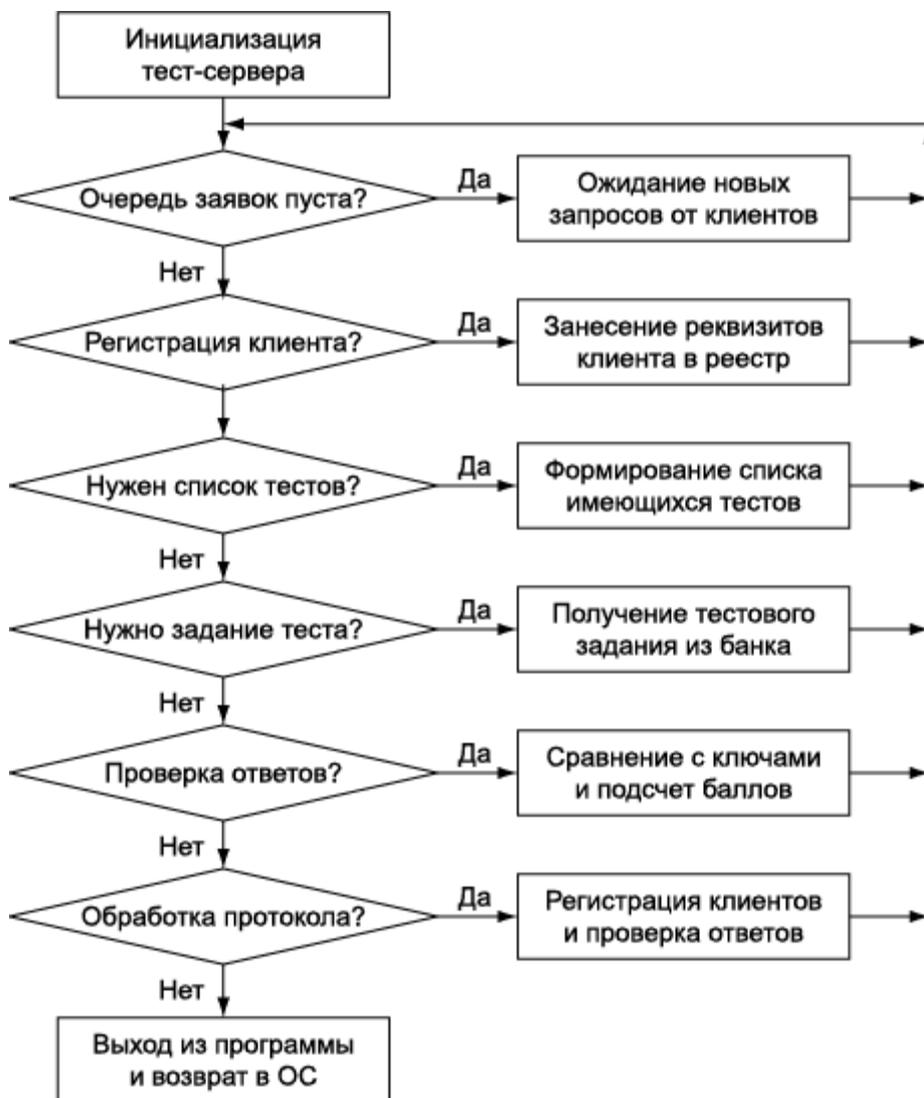
Согласно алгоритму после запуска программы и ввода пароля возможны следующие действия (большинство из них доступны только инструктору):

- добавление тестов, полученных на тест-сервере, в единый для локальной сети банк тестовых заданий;
- проведение сеанса тестирования, включая регистрацию, выбор теста, выполнение тестовых заданий (уже загруженных в локальную сеть), приостановку и завершение работы с тестом;
- формирование протоколов тестирования с реквизитами и ответами участников, отправка их через Интернет на обработку тест-серверу и получение обратно результатов проверки ответов и подсчёта баллов;
- сбор и анализ статистической информации о полученных результатах, составление сводных таблиц (о количестве участников, о распределении набранных баллов, о проценте

правильных ответов на отдельные задания теста и т.д.), формирование списков тестируемых и вывод их на печать;

- удаление лишних записей из баз данных с тестами, с информацией о сеансах тестирования, с реквизитами и результатами участников.

**Алгоритм функционирования тест-сервера, обслуживающего множество рабочих станций в рамках ИВС КТ**



Он позволяет за счёт централизации данных:

- вести общий реестр с реквизитами и результатами всех участников компьютерного тестирования, оперативно получая из него информацию;
- использовать единый банк тестовых заданий, уменьшая трудозатраты на его администрирование и снижая расходы дискового пространства;
- одновременно обеспечивать возможность оперативной и отложенной обработки ответов, тестируемых через Интернет.

Сразу после инициализации, предполагающей открытие базы данных и создание очереди заявок, тест-сервер переходит в состояние ожидания новых запросов от клиентов. Получив запрос, тест-сервер определяет его параметры и выполняет одно из следующих действий:

- занесение реквизитов клиента в общий реестр;
- формирование списка имеющихся в банке тестов;
- получение запрошенного тестового задания из банка;
- сравнение ответов тестируемого с ключами и подсчёт набранных баллов;
- приём протокола, содержащего всю информацию о нескольких сеансах компьютерного

тестирования в локальной сети, его обработку, включая регистрацию клиентов и проверку их ответов, и формирование списка с итоговыми результатами.

Алгоритм допускает псевдопараллельное обслуживание на единственном процессоре запросов от разных клиентов за счёт выполнения всех операций, требующих много времени, в асинхронном режиме. При работе с протоколом (после обработки очередной записи с информацией о сеансе тестирования) текущий запрос становится в конец очереди, из которой выбирается первая заявка. Это позволяет избежать значительного увеличения времени пребывания оперативных запросов в очереди при обслуживании больших протоколов.