

## АВТОМАТИЗИРОВАННЫЙ КОМПЛЕКС ОБРАБОТКИ ТЕЛЕМЕТРИЧЕСКОЙ ИНФОРМАЦИИ

© 2003 В. А. Кузин<sup>1</sup>, Ю. И. Атаманчук<sup>1</sup>, Н. В. Кравчук<sup>1</sup>, А. П. Шибанов<sup>2</sup>, В. А. Шибанов<sup>2</sup>

<sup>1</sup>Федеральное государственное унитарное предприятие ОКБ “Спектр”, г. Рязань

<sup>2</sup>Рязанская государственная радиотехническая академия

На базе локальной сети реализован автоматизированный комплекс обработки телеметрической информации (ТМИ). Параллельно функционирующие клиентские компьютеры испытателей подключаются к серверу обработки на основе технологии сокетов. Предложены математические методы и графовые модели для определения времени выполнения типовых заданий на обработку ТМИ. Разработана имитационная модель для анализа производительности системы обработки ТМИ.

**Введение.** Большинство новых современных объектов представляют собой сложные технические комплексы (СТК) агрегатов, механизмов, приборов, средств автоматизации, связи и т. д. Постоянное развитие технических средств приводит к увеличению объемов измерительной информации, получаемой в процессе испытаний. Создание практически любого крупного СТК связана с разработкой телеметрической системы, которая учитывает все особенности данного СТК: назначение, режимы работы, область и среду применения и т. д. В настоящее время прогресс в области программирования, развитие микропроцессорной техники, активное внедрение сетевых технологий, а также появившееся научно-методическое обеспечение проектирования телеметрических систем создали реальную базу для повышения качества проектных работ, унификации средств телеизмерений и создания средств обработки ТМИ нового поколения. Использование средств автоматизированного проектирования позволяет оптимизировать структуру системы обработки ТМИ, оценить производительность и возможности ее масштабирования при увеличении объема и сложности решаемых задач.

**Описание автоматизированного комплекса обработки ТМИ.** В данной статье рассматриваются вопросы разработки программного обеспечения, предназначенного для использования в автоматизированных комплексах обработки телеметрии (АКОТ) типа “Лагуна”, “Экспресс-анализ”, КСС ТМИ (комплекс средств сбора ТМИ). Структура типовой АКОТ приведена на рис. 1.

Поток телеметрической информации от приемно-регистрирующей станции (ПРС) поступает на *адаптер* АКОТ, который преобразует аналоговую информацию в цифровой код и передает ее в *оперативное запоминающее устройство* (ОЗУ). ОЗУ предназначено для временного хранения полученной цифровой информации от адаптера. Информация из ОЗУ передается на *формирователь кадров*.

*Формирователь потоков* – программный модуль, предназначенный для формирования потоков данных в соответствии с установленными программными клиентами заданиями: кодовые значения отдельных параметров преобразуются в физические величины с помощью соответствующего *обработчика параметров*.

Телеметрическая информация с выхода ОЗУ и формирователя потоков копируется на жестком диске или на магнитной ленте.

*Программный сервер* – модуль, предназначенный для подключения и отключения клиентов и обеспечения обмена данными через локальную сеть между компьютерами-клиентами и сервером системы. Сервер по запросу клиента формирует ответ, содержащий набор требуемых параметров или причину отказа в обслуживании.

*Программный клиент* – модуль, размещенный на компьютере-клиенте, предназначен для организации взаимодействия данного компьютера с сервером системы. Программный клиент функционально связан с *анализатором результатов* – пакетом программ, предназначенным для осуществления

окончательного машинного анализа результатов испытаний (сравнение с установленными эталонами, проверка допустимости значений и т. д.), а также для представления их в удобном формате (построение таблиц, графиков, диаграмм и т. д.).

В начале сеанса сервер ожидает подключения новых клиентов. После запуска клиентского приложения на компьютере-клиенте оно подключается к серверу. Сервер назначает ему уникальный идентификатор и передает его клиенту. Как только клиент получает сообщение о подключении, он устанавливает задание на сеанс. Сервер, получив от клиента задание на сеанс, начинает формирование набора требующихся клиенту параметров. После получения ответа на запрос клиент передает полученные параметры на дальнейшую обработку анализатору результатов.

В качестве среды разработки программного обеспечения используется Borland Delphi 5 фирмы Inprise. Создание приложений в Delphi базируется на объектно-ориентированном программировании. Для создания многопользовательских распределенных приложений в Delphi 5 предусмотрен интерфейс сокетов.

*Сокеты* (sockets – “гнезда”) – это один из способов передачи данных и обмена ин-

формацией между компьютерами. Сокеты являются программными конечными точками сетевого соединения. Для работы с сокетами необходимо использовать некоторый протокол на основе TCP/IP и программный порт транспортного уровня Windows. Сокеты делятся на три основных типа.

*Клиентские сокеты* инициализируют соединение со стороны клиента с *серверным сокетом* на удаленной машине. Для того чтобы открыть соединение, клиентский сокет должен “знать” IP-адрес удаленной машины и номер порта, используемого серверным сокетом. Клиент посылает серверу запрос на соединение. Серверные сокеты сами не занимаются установлением связи с клиентскими сокетами. Эту задачу выполняют *слушающие сокеты*, встроенные в серверные сокеты. Запрос на подключение нового клиента получает слушающий сокет, который ставит его в очередь. После того, как серверный сокет освободится от текущей работы, он обрабатывает запрос из очереди и создает слушающий сокет для нового соединения. Серверные сокеты устанавливают соединение с клиентским сокетом в ответ на его запрос. Клиентский сокет получает описание серверного сокета, после чего соединение считается установленным.

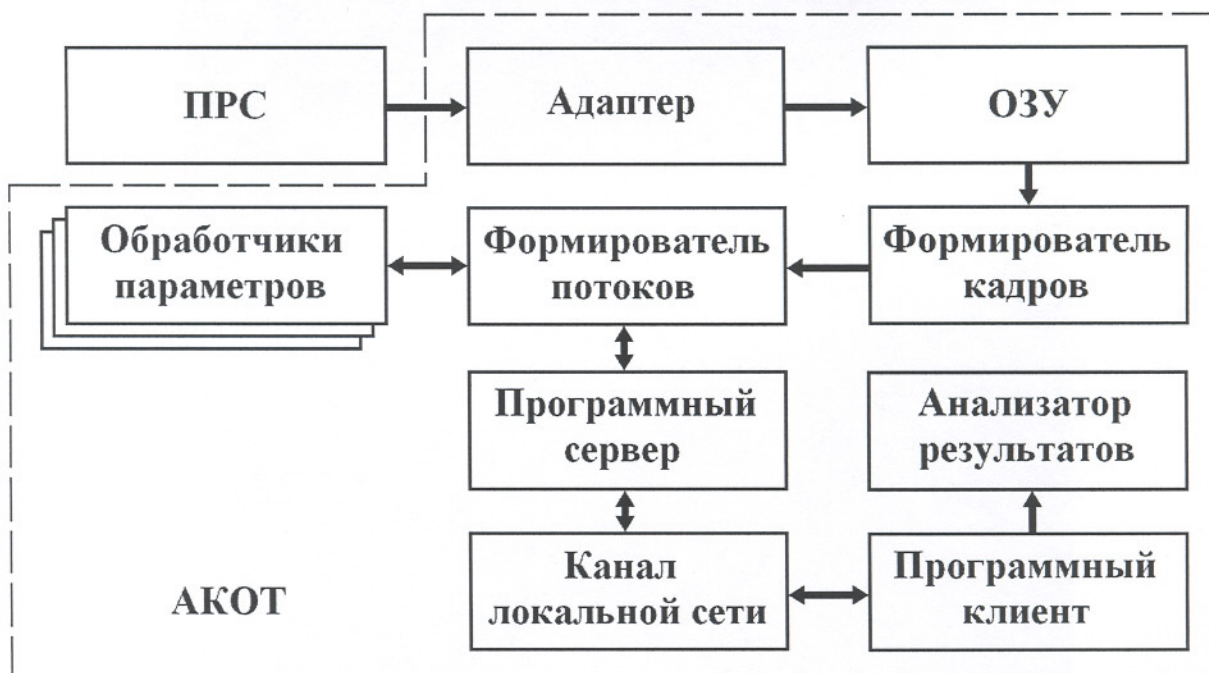


Рис. 1. Обобщенная структура АКОТ

Программные модули реализованы в виде динамически подключаемых библиотек DLL. При этом обеспечивается возможность изменения алгоритмов работы разрабатываемых модулей без внесения каких-либо изменений в остальные блоки системы. Программы взаимодействия сервера с клиентами функционируют на компьютере, имеющем следующую конфигурацию: Pentium III – 533 и выше; 64 Мбайт ОЗУ; монитор SVGA; ОС Windows 95, 98, NT.

Предполагается, что локальная сеть не вызывает задержек при передаче кадров между программным сервером и программными клиентами. Этого легко достичь, например, соединением через безынерционный коммутатор сегментов Fast Ethernet с 5-7 компьютерами в каждом или применением сети Gigabit Ethernet [1].

При высокой загрузке системы программа-сервер обрабатывает клиентские задания последовательно, без каких-либо перерывов. Совокупность поступающих на сервер заданий на обработку наборов данных образует трафик определенного профиля. Его структура существенно зависит от числа и спецификаций параметров, запрашиваемых в одном наборе данных, числа наборов данных и интервалов следования между ними. Число наборов, число параметров в наборе и время обработки параметров являются случайными величинами. При увеличении числа клиентских программ увеличивается длина буфера в ОЗУ и возрастает время обработки кадра. Одной из наиболее важных задач проектирования данной системы является определение максимально возможной производительности АКОТ. Для этого необходимо разработать методику определения предельно допустимого числа клиентских программ при известных профилях вырабатываемого ими трафика.

**Аналитические методы оценки времени выполнения заданий клиентов.** Время выполнения задания программным сервером определяем путем нахождения суммы случайного числа независимых случайных величин  $X_1, X_2, \dots$  с одним и тем же распределением  $F$  и производящей функцией

моментов  $M(s)$ . Пусть  $N$  – целочисленная случайная величина с производящей функцией  $A(s) = \sum p_i s^i$  и не зависящая от всех  $X_j$ .

Тогда случайная сумма  $X_1 + \dots + X_N$  имеет распределение, описываемое производящей функцией моментов

$$\omega(s) = A(M(s)), \quad (1)$$

где  $A(s)$  – производящая функция, описывающая случайное число запрашиваемых программой-клиентом параметров,  $M(s)$  – производящая функция моментов времени обработки одного параметра [2].

Рассмотрим метод расчета задержек при описании числа затребованных программой-клиентом параметров *равномерным распределением с целочисленными значениями*. Число параметров в задании может изменяться от  $h$  до  $l$ . Найдем производящую функцию моментов этого распределения с учетом того, что все события считаются равновероятными со значением  $\bar{p}$ :

$$\begin{aligned} M(s) &= \bar{p}(e^{sh} + e^{(h+1)s} + \dots + e^{(l-1)s} + e^{ls}) = \\ &= \bar{p} \frac{e^{sh} - e^{(l+1)s}}{1 - e^s}. \end{aligned}$$

Производящая функция этого распределения получается заменой  $e^s$  на  $s$ :

$$A(s) = \bar{p} \frac{s^h - s^{l+1}}{1 - s}.$$

Для оценки случайного времени обработки одного параметра используем равномерное непрерывное распределение с параметрами  $a$  и  $b$ . Тогда согласно формуле (1) имеем

$$\omega(s) = \frac{\bar{p} \left[ \left( \frac{e^{as} - e^{bs}}{(a-b)s} \right)^h - \left( \frac{e^{as} - e^{bs}}{(a-b)s} \right)^{l+1} \right]}{1 - \frac{e^{as} - e^{bs}}{(a-b)s}}. \quad (2)$$

Дифференцируя  $\omega(s)$  по переменной  $s$  и приравнявая в полученных выражениях величину  $s$  нулю, получаем первый и второй моменты относительно начала координат и, соответственно, среднее значение  $\tilde{t}_{cp}$  и дисперсию  $\tilde{\sigma}^2$  времени выполнения одного задания программы-клиента:

$$\mu_1 = \tilde{t}_{cp} = \left. \frac{\partial[\omega(s)]}{\partial s} \right|_{s=0} = \frac{(h+l)(a+b)}{4}, \quad (3)$$

$$\begin{aligned} \tilde{\sigma}^2 &= \left. \frac{\partial^2[\omega(s)]}{\partial s^2} \right|_{s=0} - \left( \left. \frac{\partial[\omega(s)]}{\partial s} \right|_{s=0} \right)^2 = \\ &= \frac{(h+l)(b-a)^2}{24}. \end{aligned} \quad (4)$$

Если время преобразования одного кодового значения в физическую величину описывается экспоненциальным распределением с параметром  $\lambda$ , то имеем

$$\tilde{t}_{cp} = \frac{h+l}{2\lambda}, \quad \tilde{\sigma}^2 = \frac{h+l}{2\lambda^2}. \quad (5)$$

Для нормального распределения времени обработки кодового значения с математическим ожиданием  $m$  и дисперсией  $\sigma^2$  получаем следующие характеристики:

$$\tilde{t}_{cp} = \frac{(h+l)m}{2}, \quad \tilde{\sigma}^2 = \frac{(h+l)\sigma^2}{2}. \quad (6)$$

На практике часто выполняются условия, при которых число затребованных программой-клиентом параметров может быть описано *распределением Пуассона*. В данном случае эти условия заключаются в том, что информация о числе параметров, запрошенных клиентской программой в любом предыдущем задании, ничего не говорит об их числе в данном задании. Производящая функция распределения Пуассона равна  $A(s) = \exp(\lambda s - \lambda)$ . Если время преобразования кода параметра в физическую величину описывается равномерным распределением,

то производящая функция моментов времени выполнения задания программы-клиента равна

$$\omega(s) = \exp \left[ -\lambda + \lambda \frac{e^{as} - e^{bs}}{(a-b)s} \right]. \quad (7)$$

Из выражения (7) находим среднее время выполнения задания и его дисперсию

$$\tilde{t}_{cp} = \frac{\lambda(a+b)}{2}, \quad \tilde{\sigma}^2 = \frac{\lambda(a^2 + ab + b^2)}{3}. \quad (8)$$

В большинстве случаев плотность распределения вероятностей времени выполнения одного задания  $\varphi(x)$  имеет одну моду. Формулы (3) – (6), (8) можно использовать для предварительной оценки величины разброса распределения на основе правила “трех сигм”.

Знание функции распределения времени выполнения задания  $F(x)$  необходимо для моделирования задержек в более сложной модели, основанной на компьютерной имитации. Для нахождения  $\varphi(x)$  могут быть использованы графо-аналитические GERT-модели. Они позволяют описывать функционирование системы, последовательно переходящей из одного состояния в другое:  $C_1, C_2, \dots$  (число состояний конечно или бесконечно). Каждому из них приписана некоторая вероятность  $q_k$ ; вероятности последовательности прохождения состояний определяются по правилу умножения  $P\{(C_{j_s}, C_{j_i}, \dots, C_{j_n})\} = q_{j_s} q_{j_i} q_{j_n}$ . Каждой паре  $(C_j, C_k)$  соответствует условная вероятность  $q_{jk}$ ; если состояние  $C_j$  достигнуто на некотором шаге, то вероятность перехода в состояние  $C_k$  на следующем шаге равна  $q_{jk}$ . В общем случае  $P\{(C_{j_s}, C_{j_i}, \dots, C_{j_n})\} = a_{j_s} q_{j_s j_i} \dots q_{j_{s+1} j_{s+2}} q_{j_{n-1} j_n}$ . Здесь  $a_k$  есть вероятность попадания в состояние  $C_k$  из начального состояния. Переход системы из состояния в состояние связывается с выполнением некоторой операции, описываемой случайной величиной с известным законом распределения. Случайные величины, характеризующие дуги GERT-сети, отражают время выполнения опе-

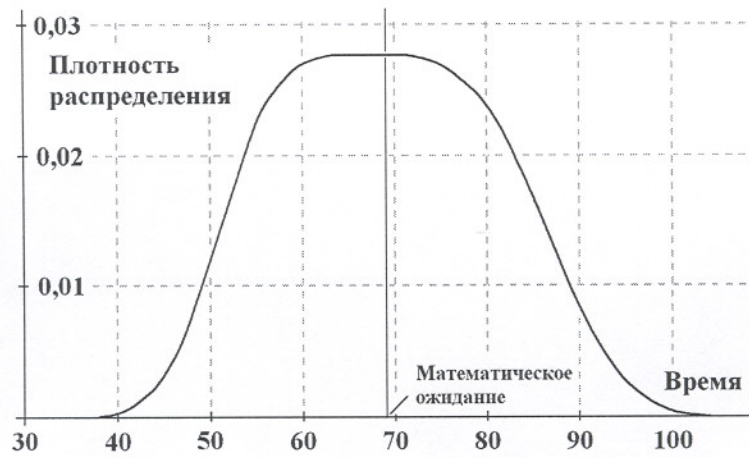


Рис. 2. Плотность распределения времени выполнения задания

раций в моделируемой системе. Метод нахождения значений  $\varphi(x)$  основан на интерполяции характеристической функции GERT-сети многочленом Лагранжа и интегрировании формулы обращения [3]. В частности, этим методом может быть получена плотность распределения вероятностей для случайных величин, описываемых выражениями (2) и (7).

Пакет прикладных программ, реализующий основные функции системы GERT, позволяет вычислить значения  $F(x)$  и записать их в табличном виде в файловую систему компьютера. При выполнении моделирования на более высоком уровне иерархии во входных спецификациях блоков, имитирующих задержки, указываются имя файла, хранящего таблицу функции распределения, и полный путь к нему. Имитационный блок реализует задержку методом обратной функции. Пример плотности распределения, полученной путем использования описанного выше метода, приведен на рис. 2. Число параметров в задании задается равномерным

распределением с целочисленными значениями от 9 до 14, а время обработки одного параметра – равномерным непрерывным распределением с параметрами  $a = 3, b = 9$ .

**Имитационное моделирование системы обработки ТМИ.** На практике часто возникают вопросы: 1) окажется ли система достаточно быстродействующей, если потребуется дополнительно обработать потоки ТМИ от нескольких объектов; 2) насколько можно увеличить количество автоматизированных рабочих мест, использующих клиентские программы, без перегрузки системы обработки ТМИ; 3) как скажется на быстродействии системы наблюдаемое в настоящее время усложнение алгоритмов обработки ТМИ. Простых расчетных методов в таких случаях может быть уже недостаточно, так как при увеличении нагрузки на комплекс обработки ТМИ важно учесть влияние многих случайных факторов, которые могут снизить его быстродействие. Проблемы такого рода эффективно решаются методом имитационного моделирования. Для исследования задерж-

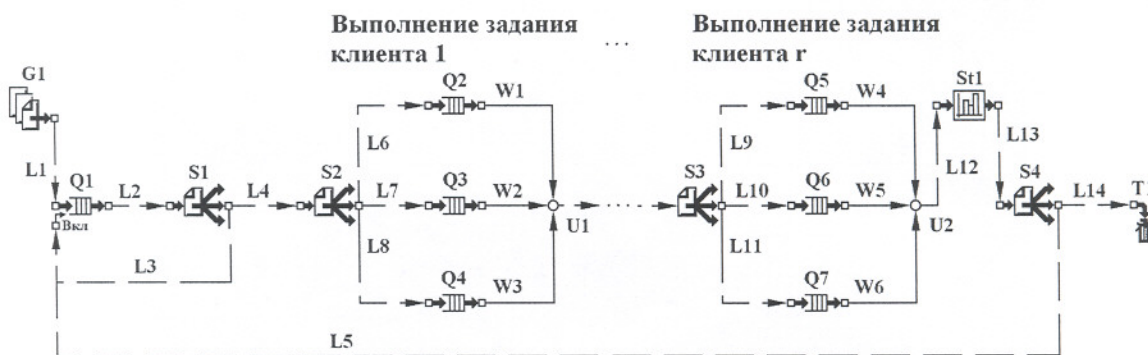


Рис. 3. Имитационная модель АКOT

жек, связанных с обработкой сервером заданий, поступающих от клиентских программ, применим программные средства, описанные в [4].

Во входной буфер системы поступают на обработку кадры с интервалом следования, определяемым темпом их поступления от ПРС. Кадр удаляется из буфера после того, как программой-сервером будут выполнены задания всех клиентских программ. Число параметров в кадрах может существенно различаться. Это зависит от типа параметров: функциональных, сигнальных одноразрядных, сигнальных многоразрядных (кодированных), диапазонных, цифровых и числа разрядов, необходимых для хранения параметра. Трафик, создаваемый клиентскими программами, моделируется путем вероятностного выбора из набора типовых заданий. Число заданий, выполняемых программой-сервером, последовательно увеличивается от прогона к прогону до значения, пока величина очереди поступающих кадров не превысит некоторое критическое значение. Имитационная модель АКOT изображена на рис. 3.

Генератор G1 вырабатывает 200 кадров с интервалом 20 мс. Блок-терминатор предназначен для проверки выполнения условий окончания моделирования или по истечении заданного модельного времени или по числу вошедших в него и уничтоженных кадров. В данном случае терминатор T1 обеспечивает окончание процесса моделирования после входа в него 200 кадров. Функции ОЗУ ото-

бражает имитационный блок Q1. Во входных спецификациях этого блока определен достаточно большой его размер (500 кадров). Обработка кадров ведется по правилу "первым пришел – первым обслужен" (First Input First Output, FIFO). Модель состоит из нескольких фрагментов, каждый из которых имитирует процесс выполнения заданий программ-клиентов  $1, \dots, r$ .

Селектор S1 выдает заявку на обработку очередного кадра через последовательную цепочку фрагментов модели, отражающих процесс выполнения заданий программ-клиентов. Одновременно с этим по дуге L3 он выдает команду на блокировку выхода очереди Q1 (вход не блокируется). Только после прохождения заявки на обработку через все фрагменты модели, отражающие выполнение заданий клиентских программ, выход очереди Q1 разблокируется командой, поступающей с выхода селектора S4 по дуге L5. Если в очереди Q1 имеется хотя бы один кадр, ожидающий обработки, то начинается новый цикл моделирования.

В первый фрагмент модели заявка на обработку подается через селектор S2, который передает ее в один из блоков-очереди Q2, Q3, Q4 в соответствии с вероятностным выбором. Каждая из этих очередей является вспомогательным буфером для выполнения операций программы-сервера с одним кадром. Дуги W1, W2, W3 реализуют случайные задержки методом вычисления обратной фун-

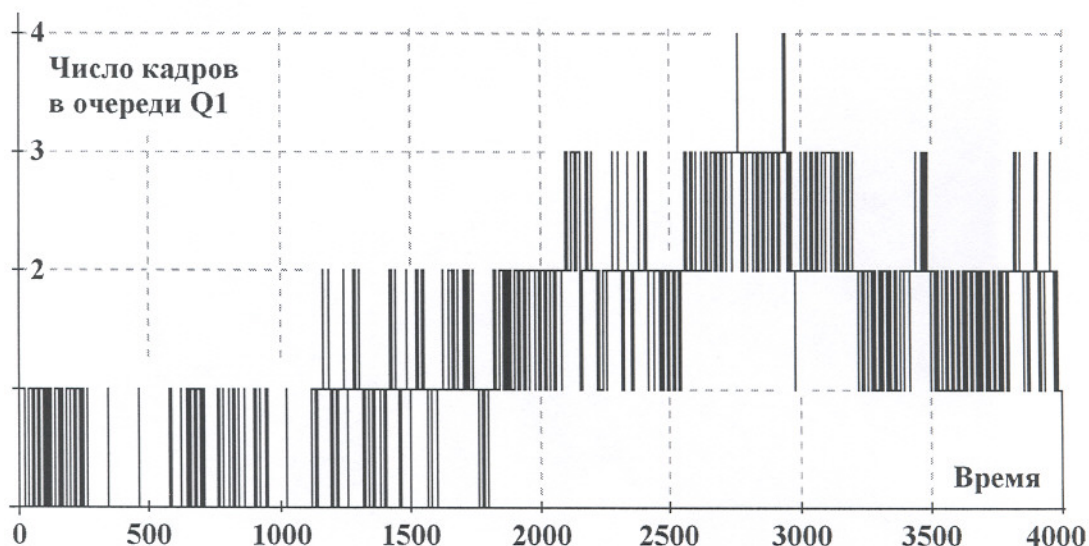


Рис. 4. Число кадров в очереди Q1 при работе 7 клиентских программ

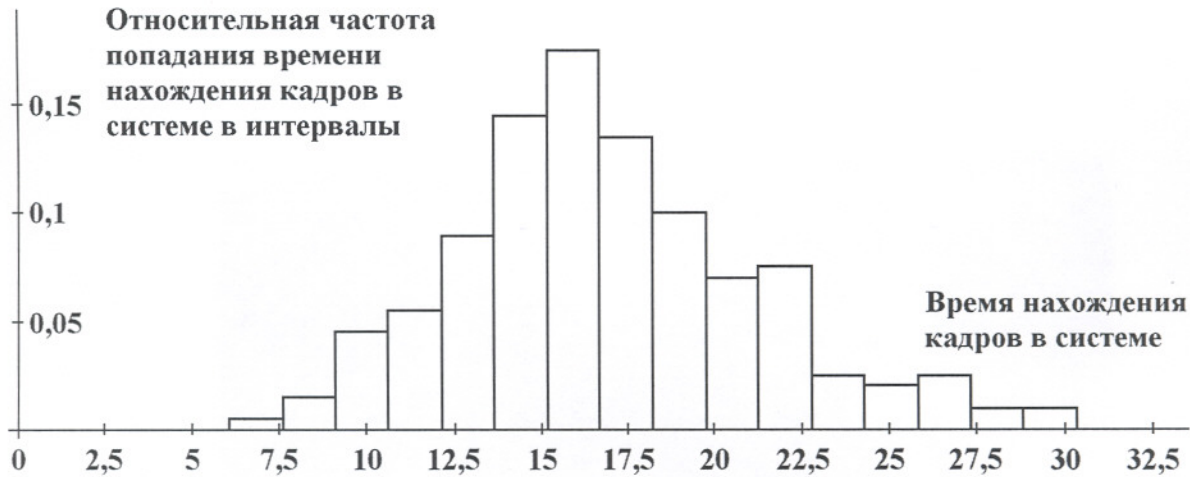


Рис. 5. Гистограмма времени обработки кадра

кции распределения времени исполнения программой-сервером одного клиентского задания. В данной модели задержка задается во входных спецификациях блока прямым указанием закона распределения. При необходимости эти данные могут считываться из файла, записанного на винчестере. Дуга W1 характеризует выполнение программой-сервером задания программы-клиента на обработку 1000 параметров со случайным временем обработки одного измерения, описываемым нормальным распределением с параметрами  $m = 5, \sigma^2 = 1$ . Дуга W2 отражает операции пересчета 500 кодовых значений в физические величины с тем же случайным временем обработки одного параметра, что и в предыдущем случае. Дуга W3 характеризует вы-

полнение задания по обработке 100 кодовых параметров ТМИ. Время обработки одного параметра задается нормальным распределением с параметрами  $m = 9, \sigma^2 = 3$ . Через узел U1 заявка на обработку передается во второй фрагмент модели, отражающий операции по обработке сервером задания второй программы-клиента и т. д.

При выполнении моделирования от прогона к прогону увеличивалась нагрузка на систему за счет увеличения числа клиентских заданий. При  $r = 7$  число кадров во входном буфере системы (очередь Q1) не превышало 4 (рис. 4) и отсутствовала тенденция к росту очереди. В модель введен блок сбора статистики St1. Такие блоки можно ставить в любой цепи модели и получать гистограм-

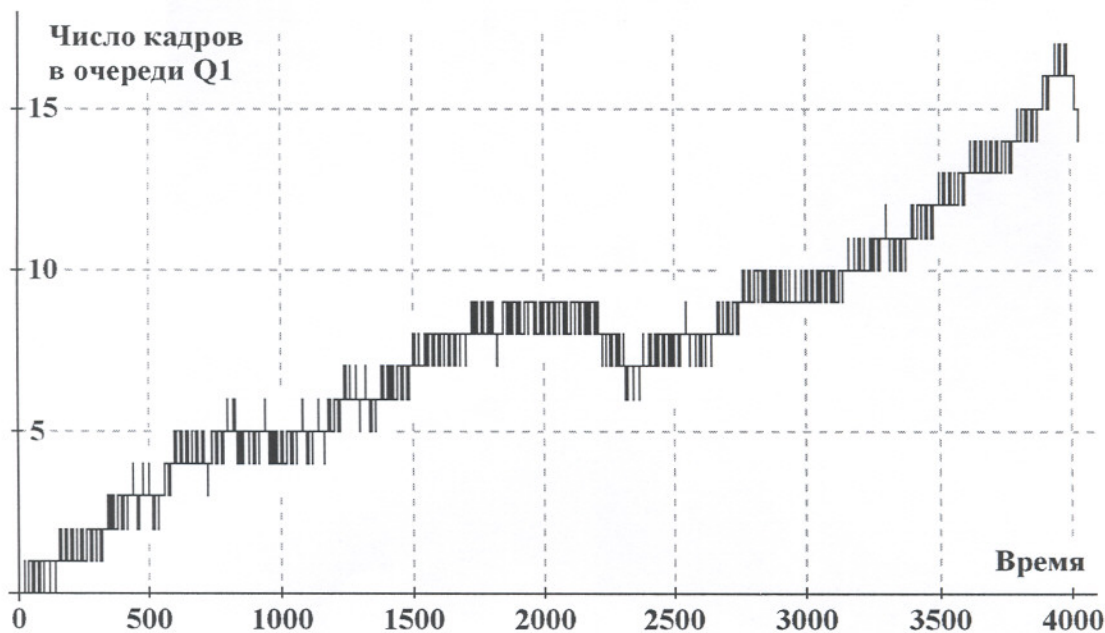


Рис. 6. Число кадров в очереди Q1 при работе 8 клиентских программ

мы: времени входов заявок в систему; времени прохождения заявки от момента ее рождения в блоке-генераторе до попадания в блок сбора статистики; интервалов между заявками. Гистограмма нахождения заявки в системе при  $r = 7$  приведена на рис. 5. Можно сделать вывод о том, что при заданных условиях функционирования АКОТ будет обладать достаточно хорошими характеристиками производительности, если он будет нагружен не более чем 7 клиентскими компьютерами.

При обслуживании сервером 8 клиентских программ наблюдается заметный рост входной очереди Q1 (рис. 6). Если увеличить время моделирования (или число вырабатываемых генератором G1 кадров), то рост этой очереди будет продолжаться. Это говорит о перегрузке системы.

**Заключение.** Использование аналитических методов моделирования, описанных в данной статье, позволяет выполнить предварительные расчеты производительности системы. Для определения максимальной производительности АКОТ целесообразно использовать графовые модели алгоритмов и программ для получения возможно более точных сведений о случайном времени выполнения программой-сервером заданий кли-

ентских программ. Полученные функции распределения используются в системе имитационного моделирования.

Применение разработанных методов и моделей целесообразно как на ранних стадиях проектирования системы, так и во время ее промышленной эксплуатации для определения предельно допустимого числа одновременно работающих клиентских программ с учетом структуры заданий, частоты и распределения во времени запросов на обработку ТМИ к программе-серверу.

#### Список литературы

1. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы. СПб: Питер, 2001. 672 с.
2. Феллер В. Введение в теорию вероятностей и ее приложения. В 2-х томах. Т. 2. М.: Мир, 1984. 738 с.
3. Корячко В. П., Шибанов А. П., Шибанов В. А. Численный метод нахождения закона распределения выходной величины GERT-сети // Информационные технологии, № 7. 2001. С. 16 – 21.
4. Шибанов А. П. Моделирование работы коммутатора в полнодуплексном режиме // Вестник СГАУ. Самара, 2002. № 1. С. 143-150.

## AUTOMATED COMPLEX OF PROCESSING TELEMETRY INFORMATION

© 2003 V. A. Kuzin, Y. I. Atamanchuk, N. V. Kravchuk, A. P. Shibanov, V. A. Shibanov

<sup>1</sup>Federal State Unitary Enterprise "Spektr", Ryazan

<sup>2</sup>Ryazan State Radio Engineering Academy

Automated complex of processing telemetry information is realized on the basis of a local network. Customer computers of the testers functioning in parallel are connected to the processing server on the basis of socket technology. Mathematical methods and graphic models for defining the time of executing standard jobs of telemetry information processing are proposed. A simulation model for the analysis of telemetry information processing system productivity has been developed.