

Ю.С. Данилюк, Ф.А. Попов

**Подход к созданию имитационных моделей компьютерных сетей**

В статье рассматриваются вопросы, связанные с созданием имитационных моделей современных локальных и корпоративных вычислительных сетей, ядром которых являются коммутаторы пакетов Ethernet, определяющие в значительной мере такие сетевые характеристики, как производительность, перегрузочная способность и отказоустойчивость.

Разработанная для этих целей *среда имитационного моделирования вычислительных сетей* позволяет детально исследовать процессы, протекающие в компьютерных сетях при их создании и последующей эксплуатации [1]. Для построения моделей в данной среде предусмотрено наличие связанных групп объектов, представляющих те или иные элементы реальной сети. Имитация процессов обмена информацией между объектами модели осуществляется путем последовательной обработки событий, каждое из которых отражает проявление тех или иных процессов в элементах исследуемой сети. Обработка события производится методами

объекта, который непосредственно затрагивается событием и может приводить к порождению новых событий или изменению свойств взаимосвязанных с ним объектов. События в рамках среды моделирования реализуются с помощью механизма сообщений, которыми обмениваются объекты модели. В процессе обработки сообщений происходит изменение состояния объекта и расчет показателей, характеризующих его работу. Рисунок 1 иллюстрирует работу модели сети по доставке сообщения, состоящей из двух ЭВМ, объединенных коммутатором. В настоящее время в среде моделирования предусмотрено семь классов объектов:

- сетевое ПО – имитируют работу прикладного программного обеспечения на нескольких ЭВМ;
- ЭВМ – имитируют работу ЭВМ и некоторых элементов операционной системы;
- стеки протоколов – отражают работу некоторых элементов реализации стека протоколов TCP/IP в операционной системе;

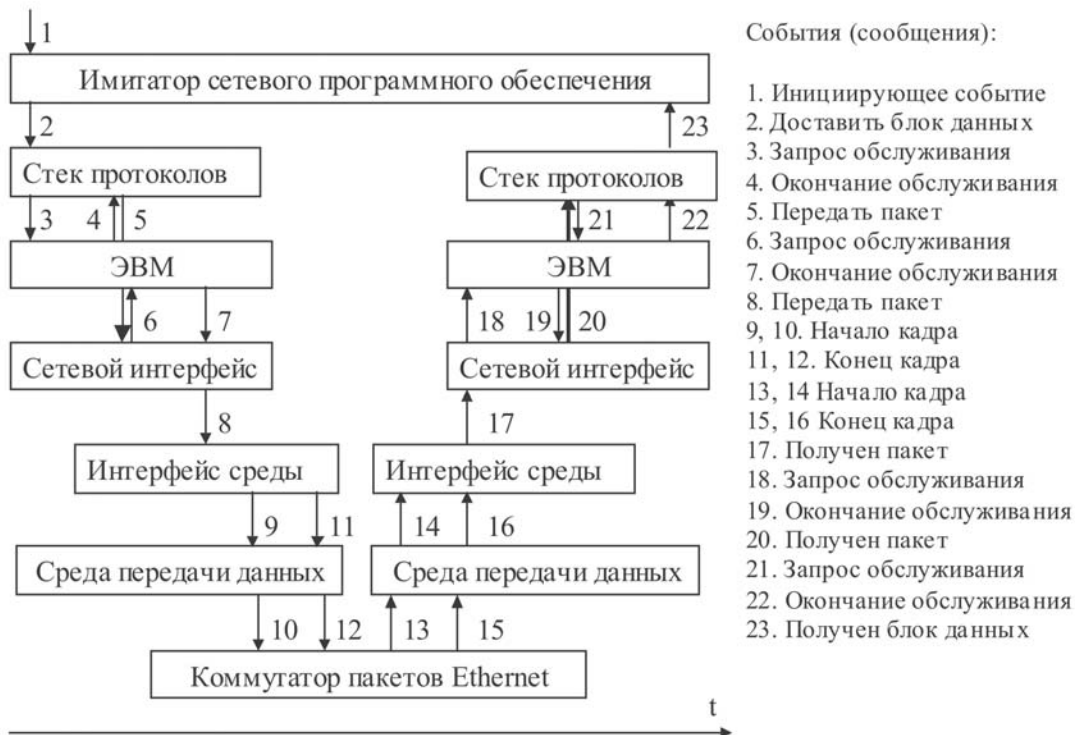


Рис. 1. Иллюстрация работы модели сети ЭВМ



Сообщения (события):

$s_1$  – начало кадра;  $s_2$  – конец кадра;  $s_3$  – начало кадра;  $s_4$  – конец кадра.

Рис. 2. Иллюстрация работы модели коммутатора

- сетевые интерфейсы – имитируют работу сетевых интерфейсов ЭВМ и системного ПО, обслуживающего интерфейс (драйверов);
- интерфейсы среды – имитируют работу приемо-передающей части сетевого оборудования;
- среды передачи данных – имитируют задержки, вносимые реальной физической средой передачи данных;
- коммутаторы пакетов Ethernet – моделируют работу сетевых коммутаторов.

Задача построения модели легко решается с помощью инструментального комплекса, автоматически генерирующего программный код исходя из описания модели, построенного в специальном редакторе. В процессе моделирования основной программный модуль производит последовательный вызов методов объектов, осуществляющих выборку значений показателей, характеризующих работу интересующих элементов сети.

**Модели коммутаторов сетей Ethernet.** Одной из основных характеристик вычислительных сетей является их производительность, зависящая в основном от используемых сетевых коммутаторов. Работа модели коммутатора в рамках рассматриваемой методики проиллюстрирована на рисунке 2, где по оси времени отложена последовательность сообщений (событий)  $\{s_i\}$ , вызывающих изменения состояния модели  $z_i$  в момент времени  $t_i$ . Промежутки времени  $t_{1,2}$  и  $t_{3,4}$  отражают время передачи кадра в средах передачи данных 1 и 2 соответственно, а промежуток времени  $t_{2,3}$  – задержку коммутации пакета  $t$ .

При этом модель коммутатора должна адекватно воспроизводить задержки коммутации, потери пакетов, управление интерфейсами, приоритеты трафика и другие эффекты, связанные с передачей информации, присущие моделируемому коммутатору [2].

Процесс создания модели коммутатора состоит из четырех этапов:

- построение тестовой ЛВС;
- разработка алгоритма работы модели, обеспечивающего воспроизведение требуемых особенностей передачи пакетов коммутатором;
- определение значений параметров разработанного алгоритма модели;
- проверка адекватности и уточнение модели.

**Построение тестовой ЛВС.** В общем виде тестовая ЛВС состоит из нескольких компьютеров, оснащенных сетевыми интерфейсами, непосредственно подключенными к портам исследуемого коммутатора. Входящие в состав тестовой ЛВС компьютеры должны обладать операционной средой, совместимой с тестовым программным обеспечением, а также достаточной для проведения испытаний производительностью. Максимальная битовая скорость сетевого интерфейса должна быть выше или соответствовать таковой у исследуемого коммутатора. Число компьютеров в идеальном случае должно соответствовать числу портов исследуемого коммутатора, что позволит получать исчерпывающую информацию о его работе.

**Разработка алгоритма работы модели.** Предусматриваются два вида алгоритмов: основанных на реализации некоторого набора аналитических выражений и эмулирующих процессы, протекающие в моделируемых устройствах.

Пример, иллюстрирующий использование алгоритма, основанного на реализации набора аналитических выражений, представлен на рисунке 3. В данном примере работа модели описывается булевой функцией  $F_1()$  и функцией  $F_2()$ , которые определяют вероятность потери коммутатором поступившего пакета и задержку коммутации соответственно.



Сообщения (события):  $s_1$  – конец кадра;  $s_2$  – начало кадра.

Условия коммутации:  $P$  – вероятность потери пакета,  $\tau$  – задержка.

Рис. 3. Иллюстрация работы алгоритма первого вида

Работа алгоритма эмулирующего процессы, протекающие в моделируемых устройствах, проиллюстрирована на рисунке 4. В данном примере модель коммутатора реализуется набором нескольких объектов, эмулирующих работу блоков моделируемого устройства. Потери пакетов в такой модели будут определяться функционированием объекта «буфер», а задержка коммутации – функционированием объектов «управляющее устройство» и «порт» и по формуле:

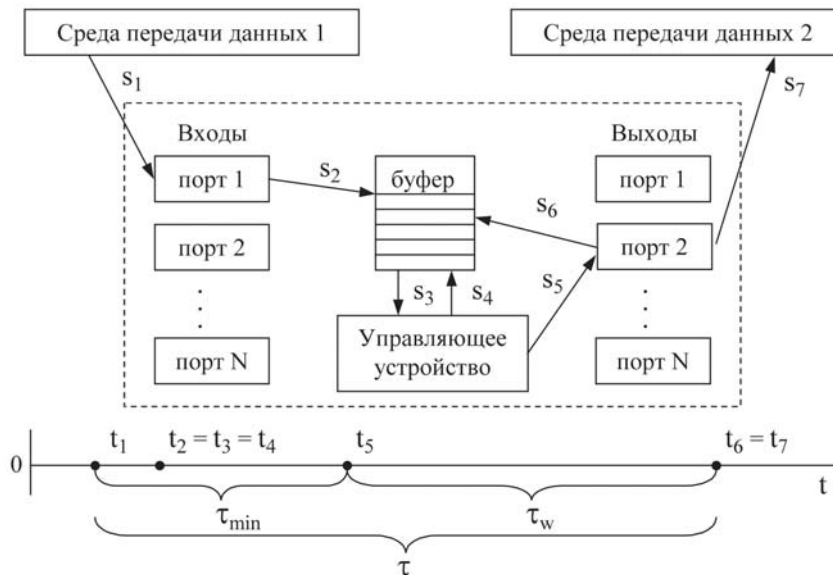
$$\tau = \tau_{\min} + \tau_w,$$

где  $\tau_{\min}$  – минимальная задержка коммутации,  $\tau_w$  – задержка, связанная с передачей объектом «порт» уже находящегося в буфере пакетов.

К достоинствам алгоритмов первого вида следует отнести их высокую вычислительную

эффективность и относительную простоту процедур определения параметров. При этом они обладают рядом недостатков, таких как непригодность моделей за пределами диапазона испытанных состояний; существенная зависимость точности результатов от объема проводимых испытаний; усложнение алгоритмов и многократное увеличение объемов испытаний в случае моделирования коммутатора, реализующего несколько механизмов классификации и дисциплин обслуживания пакетов [3]. В целом указанные особенности данных алгоритмов делают их практически непригодными для создания моделей коммутаторов, реализующих развитые средства управления трафиком.

Алгоритмы, основанные на методах эмуляции, позволяют создавать модели сложных



Сообщения (события) в модели:  $s_1$  – конец кадра,  $s_2$  – размещение пакета в буфере,  $s_3$  – сигнализация об успешном размещении пакета,  $s_4$  – определение числа операций копирования из буфера для пакета,  $s_5$  – наличие пакета,  $s_6$  – пакет скопирован,  $s_7$  – начало кадра.

Рис. 4. Иллюстрация работы алгоритма второго вида

управляемых коммутаторов, реализующих практически все виды средств управления трафиком на различных уровнях их детализации. Недостатками этих алгоритмов являются их низкая вычислительная эффективность и необходимость наличия детальных сведений о внутреннем устройстве и работе моделируемого коммутатора.

Выбор типа алгоритма, подбор и адаптация к новым особенностям функционирования подходящего алгоритма из имеющихся наработок, при необходимости – синтез нового алгоритма выполняются на основе требований к модели и знаний о структуре моделируемого устройства.

**Определение значений параметров алгоритма модели.** После выбора, адаптации или синтеза алгоритма в тестовой ЛВС выполняется серия экспериментов, направленных на определение его параметров путем статистической обработки результатов передачи через исследуемый коммутатор потоков пакетов с заданными свойствами. Генерацию исходящих и регистрацию входящих потоков пакетов выполняет тестовое программное обеспечение, в функции которого также входит настройка параметров сетевых интерфейсов; заполнение адресной таблицы коммутатора; отслеживание и управление состоянием сетевых интерфейсов в процессе испытания (контроль механизмов управления потоком); синхронизация времени на ЭВМ, участвующих в испытании; сбор результатов испытаний и перевод их в единую систему координат времени.

Рассмотрим приемы определения параметров на примере модели эмуляции неуправляемого неблокирующего коммутатора, построенного на основе архитектуры с разделяемой буферной памятью. Рисунке 5 иллюстрирует определение минимальной задержки коммутации, в котором производятся измерения времени, затрачиваемого на передачу пакета в сети из двух ЭВМ. Измерения осуществляются по схеме сети с коммутатором (схема 1) и схеме с непосредственным соединением (схема 2). В приведенном рисунке  $t_1$  и  $t_2$  обознача-

ют время обращения пакета в схемах 1 и 2 соответственно;  $\tau_3$  – задержку обработки пакета в ЭВМ2;  $\tau_1$ ,  $\tau_2$  и  $\tau_4$  – задержки распространения сигнала в физической среде передачи данных, которые будут определяться по формуле:

$$\tau = \frac{n}{V_b} + \frac{l}{V_c},$$

где  $n$  – число бит в передаваемом кадре;  $l$  – длина кабеля;  $V_b$  – номинальная битовая скорость интерфейса среды;  $V_c$  – скорость распространения сигнала в физической среде.

Для рассматриваемого коммутатора время задержки коммутации будет определяться суммой минимального времени задержки коммутации и времени пребывания пакета в буферной памяти. И в случае с одним потоком, интенсивность которого меньше или равна пропускной способности интерфейса, время пребывания пакета в буферной памяти будет минимальным. Таким образом, определив предварительно задержки распространения сигнала в среде, из полученных значений времени обращения пакета для схем 1 и 2 вычисляют минимальную задержку коммутации:

$$\tau_{\min} = \frac{t_1 - 2\tau_1 - 2\tau_2 - t_2 + 2\tau_4}{2}.$$

где  $t_1$  и  $t_2$  – время обращения пакета в схемах 1 и 2 соответственно;  $\tau_1$ ,  $\tau_2$  и  $\tau_3$  будут определяться по формуле  $\tau = nV_b + \tau_{cp}$ , где  $n$  – число бит в передаваемом пакете,  $V_b$  – номинальная битовая скорость сетевого интерфейса,  $\tau_{cp}$  – задержка распространения сигнала в среде.

На рисунке 6 продемонстрирован прием определения параметров буферной памяти – суммарного объема и организации (структуры). В данном примере два потока пакетов фиксированного размера от ЭВМ1 и ЭВМ2, обозначенные на рисунке как  $\lambda_1$  и  $\lambda_2$  с интенсивностями  $I_1$  и  $I_2$  соответственно, направляются через коммутатор к ЭВМ3, образуя результирующий поток  $\lambda_3$  с интенсивностью  $I_3$ . Суть приема заключается в том, что, когда суммарная интенсивность потоков  $\lambda_1$  и  $\lambda_2$  превысит максимальную интенсивность  $I_{\max}$  для

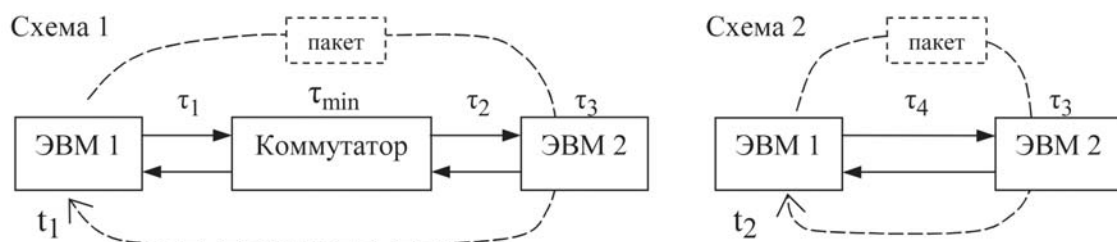


Рис. 5. Определение минимальной задержки коммутации

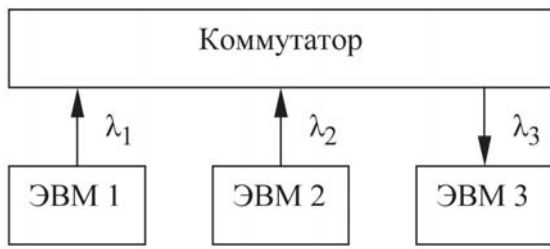


Рис. 6. Определение параметров буферной памяти

порта коммутатора, интенсивность потока  $\lambda_3$  будет равна  $I_{\max}$ , что приведет к заполнению буферной памяти коммутатора с интенсивностью  $I = I_1 + I_2 - I_{\max}$ . Когда буферная память будет заполнена, в потоке  $\lambda_3$  возникнут потери пакетов. Таким образом, момент времени, в который будет зафиксирована потеря первого пакета, будет зависеть от интенсивностей потоков и объема буферной памяти. Если потоки  $\lambda_1$  и  $\lambda_2$  запускать одновременно, выбрав  $I_1$  и  $I_2$  равными  $I_{\max}$ , и маркировать в этих потоках пакеты, то количество пакетов, уместяющихся в буфере, будет определяться как:

$$N_0 = \frac{N_1 + N_2 - 1}{2},$$

где  $N_1$  и  $N_2$  – число пакетов потоков  $\lambda_1$  и  $\lambda_2$  соответственно, принятых ЭВМ3 до регистрации первой потери пакета в потоке  $\lambda_3$ .

Организация буферной памяти, а именно эффективный размер ее страниц, в свою очередь будет определять эффективный объем буферной памяти для разных испытаний с отличающимися размерами пакетов. Для проведения этого эксперимента порты коммутатора и соответствующие сетевые интерфейсы переводятся в режим с минимально возможной номинальной битовой скоростью. Это позволяет увеличить масштаб времени и повысить точность измерений. При проведении таких испытаний в сетевых интерфейсах ЭВМ отключается реакция на кадры приостанова, генерируемые коммутатором в момент переполнения буфера. Само появление кадров приостанова в испытаниях регистрируется, а момент его появления увязывается с объемом используемой буферной памяти. Полученная величина используется в качестве параметра для модели коммутатора.

**Проверка адекватности и уточнение полученной модели.** Для этой цели в тестовой ЛВС подготавливаются и проводятся измерения, в рамках которых через коммутатор передаются и регистрируются некоторые последовательности пакетов. Затем проводится

численный эксперимент, для чего в среде моделирования создается модель тестовой ЛВС, в которой повторяются измерения, проведенные при работе с моделируемым устройством.

Подготовка к измерениям заключается в задании последовательности параметров  $t_i, n_i, m_i, s_i$ , определяющих время направления пакета в коммутатор, порт источника, порт приемника и размер передаваемого пакета соответственно.

Результаты измерений позволяют определить для каждой совокупности параметров задержки передачи пакетов  $\tau$  и  $\hat{\tau}$  в тестовой ЛВС и в среде моделирования соответственно. Путем неоднократного повторения вышеописанной процедуры измерений и проведения сравнений полученных результатов разработчиком реализуется эвристический подход к оценке адекватности и уточнению модели.

Также определенный интерес могут представлять различные реализации показателя:

$$\epsilon_\tau = \sum_{i=1}^N \left\| \tau(i) - \hat{\tau}(i) \right\|,$$

где  $\tau(i)$  и  $\hat{\tau}(i)$   $i=(1,2,\dots,N)$  – векторы задержек, образуемые одинаковыми выборками значений из массивов  $\tau[t,n,m,s]$  и  $\hat{\tau}[t,n,m,s]$  соответственно.

Данный количественный показатель, характеризующий степень различия модели и моделируемого устройства, может трактоваться как точность модели. Большой практический интерес имеет функция  $\epsilon_\tau(t)$ , представляющая изменения показателя точности модели во времени. Перед вычислением данного показателя в большинстве случаев потребуется заменить все бесконечные значения массивов  $\tau[t,n,m,s]$  и  $\hat{\tau}[t,n,m,s]$ , соответствующие наблюдаемым потерям пакетов, на одинаковые (большие) конечные величины.

При проверке моделей коммутаторов, реализующих механизмы классификации и управления трафиком, в вышеприведенные выражения, определяющие условия и результаты измерений, следует добавлять дополнительную размерность  $q$ , определяющую класс пакета, соответствующим образом скорректировав программу измерений.

Для оценки адекватности полученной модели, помимо эвристического подхода, связанного со сравнением полученных реализаций  $\tau[t,n,m,s]$  и  $\hat{\tau}[t,n,m,s]$ , можно использовать методы, основанные на статистическом подходе,

такие как, например, критерий Фишера-Снедекора, коэффициенты ранговой корреляции Спирмена и Кендела. Однако следует все же с осторожностью относиться к результатам проверки адекватности на основе статистического подхода в отношении дискретных моделей, обладающих пространством состояний. Подобные проверки должны быть подтверждены результатами практического использования полученной модели.

Рассмотренная методика моделирования вычислительных сетей позволяет эффективно решать задачи, связанные с их проектирова-

нием и оптимизацией. При этом эффективность обуславливается высокой степенью детализации моделируемых процессов, что позволяет всесторонне изучить влияние тех или иных факторов на работу сети. Помимо решения указанных задач, интерес представляет применение имитационных моделей для целей управления сетями. В этом плане в настоящее время разработаны и опробованы процедуры создания систем автоматического управления элементами вычислительной сети, использующие данные, получаемые с помощью соответствующих имитационных моделей.

### Литература

1. Данилюк Ю.С., Попов Ф.А., Максимов А.В. Моделирование локальных вычислительных сетей // Известия АГУ. 2002. №1.
2. Советов Б.Я., Яковлев С.А. Моделирование систем. М., 2001. №1.

3. Жожикашвили А.В., Вишнеvский В.М. Сети массового обслуживания. Теория и применение к сетям ЭВМ. М., 1988.