

УДК 004.021:519.6

DOI: 10.22363/2312-9735-2018-26-4-371-382

## Вычислительная и имитационная модели системы с управлением на Modelica

А. М. Ю. Апреутсей\*, А. В. Завозина\*,  
А. В. Королькова\*, Д. С. Кулябов\*†

\* *Кафедра прикладной информатики и теории вероятностей  
Российский университет дружбы народов  
ул. Миклухо-Маклая, д. 6, Москва, Россия, 117198*

† *Лаборатория информационных технологий  
Объединённый институт ядерных исследований  
ул. Жолио-Кюри, д. 6, г. Дубна, Московская область, Россия, 141980*

При моделировании сетевых протоколов является проблемой выбор модельного подхода и средства программной реализации. Специфика данной предметной области состоит в том, что для описания протоколов обычно используют дискретно-событийный подход. Однако дискретный модельный подход имеет ряд недостатков. Он плохо масштабируем, недостаточно хорошо подходит для описания динамических систем. Как альтернативу дискретному подходу обычно рассматривают непрерывный подход. Но при моделировании дискретных событий непрерывное описание становится излишне сложным и тяжеловесным. События принимают форму некоторых ограничений на непрерывную систему, которые зачастую не входят явно в непрерывную модель, а имеют форму дополнительных семантических описаний. Авторы предлагают использовать при моделировании подобных систем гибридный (непрерывно-дискретный) подход. В рамках гибридного подхода дискретная система записывается в непрерывном виде, а события принимают вид присущих подходу дискретных переходов. Кроме того, если брать за основу именно описание событий, на основе гибридного подхода можно получить и имитационную модель.

В работе демонстрируется применение гибридного подхода для описания системы с управлением на примере взаимодействия протокола TCP и алгоритма RED. Демонстрируется простота создания как вычислительной, так и имитационных моделей системы. В качестве языка реализации используется язык Modelica.

**Ключевые слова:** активное управление трафиком, имитационное моделирование, гибридное моделирование, Modelica, Random Early Detection (RED)

### 1. Введение

При моделировании сложных систем возникает проблема выбора модельного подхода. Приведём следующий пример.

Пусть имеется система управления посредством алгоритма случайного раннего обнаружения (Random Early Detection, RED) [1] потоком трафика, передаваемого по протоколу TCP (Transmission Control Protocol). В такой системе можно выделить следующие особенности.

Во-первых, модель передачи данных по протоколу TCP можно описывать как в дискретно-событийной парадигме (как это реализовано в эталонном средстве имитационного моделирования сетевых протоколов ns-2 [2, 3], но в этом случае возникает проблема масштабируемости системы), так и с использованием непрерывного подхода (представление потока данных как потока жидкости [4, 5], в этом случае усложняется математическая модель системы).

Во-вторых, модель модуля активного управления TCP-подобным трафиком может быть представлена особого вида функцией, зависящей от типа RED-алгоритма

(функция может быть кусочно непрерывной с разрывами первого рода, иметь различные параметры, влияющие на её вид и т.п.) [1, 6–8], т.е. тоже встаёт вопрос о выборе способа моделирования.

В-третьих, модель взаимодействия входящего TCP-потока и маршрутизатора, обрабатывающего поток по алгоритму типа RED, может иметь ряд ограничений, затрудняющих её исследование в рамках только одного подхода — непрерывного или дискретного.

Таким образом, встаёт проблема моделирования как протокола TCP и управляющего модуля по отдельности, так и их взаимодействия. Кроме того, необходимо выбрать адекватную методику моделирования [9–11].

Для решения проблемы дискретного и непрерывного подходов при моделировании сложных систем предлагается использовать гибридный подход [12–17]. В качестве программного средства моделирования предлагается использовать язык Modelica [18, 19], а конкретно его реализацию OpenModelica. Язык Modelica [18, 19] разработан некоммерческой организацией Modelica. Также эта ассоциация разрабатывает свободную стандартную библиотеку для этого языка. Modelica поддерживает непрерывную и гибридную (непрерывно-дискретную) парадигмы. Впрочем, чисто дискретные элементы в языке тоже присутствуют.

Структура работы следующая. В разделе 2 приводятся аргументы в пользу применения гибридного подхода к моделированию сложных систем. Затем в разделе 3 рассматривается идеология построения имитационной модели функционирования протокола TCP на языке Modelica. Показано, что по описанию в соответствии со стандартом можно получить имитационную модель функционирования данного протокола на языке Modelica. В следующем разделе 4 продемонстрированы возможности языка Modelica для численного моделирования гибридной по структуре системы взаимодействия входящего TCP-потока и маршрутизатора, обрабатывающего поток по алгоритму типа RED. В заключении делается вывод о применимости гибридного подхода к задачам моделирования сетевых протоколов.

## 2. Гибридный подход к моделированию

Гибридный подход к моделированию систем [12–16] позволяет учесть как непрерывный, так и дискретный аспекты поведения моделируемого объекта.

Гибридность поведения моделируемой системы может проявляться по-разному:

- взаимодействие объекта, описываемого непрерывной моделью, с объектом, описываемым дискретной моделью (пример — система автоматического управления, в которой модель объекта управления является непрерывной, а модель устройства управления — дискретной);
- изменение состава моделируемой системы (пример — системы с переменным числом компонентов);
- скачкообразные качественные изменения состояний непрерывной модели объекта (в качестве дискретных событий выступают моменты качественного изменения поведения непрерывной модели).

При гибридном подходе к моделированию акцент можно делать или на дискретности исходной системы и учитывать длительность входных и выходных действий, или на непрерывности исходной системы с допущением наличия мгновенных событий наряду с длительными по времени.

Рассмотрим второй случай и будем добавлять к существующей динамической модели дискретные элементы. В качестве таких элементов могут выступать:

- начальные условия или скачкообразное изменение параметров системы в правой части дифференциального уравнения;
- число уравнений, описывающих поведение динамической системы;
- форма правых частей дифференциальных уравнений.

В рамках гибридной модели естественным образом можно представить как индикаторные функции, так и дифференциальные включения, так как скачкообразное

изменение параметров можно представить как изменение начальных условий в модифицированной системе дифференциальных уравнений. Этот приём позволяет заменить систему с изменяющейся правой частью на систему с постоянной правой частью, но изменяющимися начальными условиями.

Например, система

$$\frac{dx}{dt} = f(x, t, \pi), \quad \pi = \begin{cases} \psi_1, & x \in \mathfrak{X}_1, \\ \psi_2, & x \in \mathfrak{X}_2, \end{cases} \quad (1)$$

в которой  $\pi$  — кусочно-постоянный параметр, может быть представлена в виде

$$\begin{cases} \frac{dx}{dt} = f(x, t, \pi), \\ \frac{d\pi}{dt} = 0 \end{cases} \quad (2)$$

с начальными условиями

$$\begin{cases} \pi(0) = \psi_1, & x \in \mathfrak{X}_1, \\ \pi(0) = \psi_2, & x \in \mathfrak{X}_2. \end{cases} \quad (3)$$

Такой приём возможно применить при моделировании поведения протокола TCP и механизма RED в разных состояниях.

### 3. Имитационное моделирование функционирования протокола TCP на языке Modelica

Обычно функционирование сетевых протоколов формулируется в форме дискретно-событийного описания. Рассмотрим идеологию построения имитационной модели функционирования протокола TCP на языке Modelica.

Есть несколько стандартов протокола TCP, отличающихся реализацией механизма скользящего окна, применяемого для управления перегрузками в сетях передачи данных. В данной работе будем моделировать протокол TCP Reno [20, 21], поскольку именно на нём базировалась оригинальная модель [22–25], используемая в наших исследованиях в качестве базовой.

Особенностью протокола TCP Reno является зависимость изменения размера окна перегрузки (Congestion Window, CWND) от той или иной фазы механизма управления перегрузками. Всего таких фаз четыре: медленный старт, предотвращение перегрузок, быстрое восстановление, тайм-аут. Работа алгоритма предотвращения перегрузок в TCP описывается в RFC 5681 [21].

В фазе медленного старта окно перегрузки растёт линейно  $cwnd = cwnd + 1$  в случае прихода на источник сообщения, что оповещенный пакет доставлен (Acknowledge, ACK). Первоначальный размер окна перегрузки (Minimum Segment Size, MSS) может принимать значение 1, 2 или 10 сегментов. Сообщения о доставке отправляются приёмником для каждого пакета. Для упрощения можно предположить, что по истечении времени двойного оборота (Round-Trip Time, RTT) источник получает сразу все подтверждения о доставке отправленных пакетов. В этом случае можно считать, что по истечении времени двойного оборота происходит удвоение размера окна перегрузки.

Переход в фазу предотвращения перегрузки происходит при достижении размером окна TCP определённого размера. Меняется и схема изменения размера окна — для каждого подтверждения ACK окно перегрузки увеличивается на величину  $1/cwnd$ . Это эквивалентно увеличению окна на один пакет за время двойного оборота.

Переход в фазу быстрого восстановления происходит при получении сообщения типа тройного дублирования подтверждения (Triple Duplicate ACK, TD ACK). В этой фазе размер окна уменьшается в 2 раза  $cwnd \rightarrow cwnd/2$ .

Фаза тайм-аут (Timeout, TO) возникает, когда источник не получает подтверждения о доставке пакета в течение заданного времени. В этом случае размер окна перегрузки устанавливается в начальное значение и протокол переходит в фазу медленного старта.

Важную роль в работе протокола TCP играет *таймер повторной передачи* (Retransmission timer). Значение этого таймера инициализируется при отправке сегмента данных. Подтверждение о доставке отправленного сегмента должно быть получено до срабатывания таймера. Иначе отправка сегмента данных должна быть повторена. При этом значение таймера остаётся постоянным в фазах медленного старта и избегания перегрузок, а в фазах быстрого восстановления и тайм-аута таймер будет линейно уменьшаться с течением времени.

Опишем переходы между состояниями TCP.

Переменная *ssth* используется для перехода от первоначальной фазы медленного старта протокола TCP к фазе предотвращения перегрузки. Инициализация этой переменной происходит при открытии соединения протокола TCP путём задания максимально возможного размера окна передачи. Собственно переход от фазы медленного старта к фазе предотвращения перегрузок происходит при достижении *cwnd* заданного размера *ssth*.

При потере пакетов происходит переход либо в состояние быстрого восстановления, либо в состояние тайм-аута. Данный переход мы моделируем эмпирически в зависимости от порогового размера окна (*timeout\_th*) [26].

Аналогично из состояния избегания перегрузки можно перейти либо в состояние быстрого восстановления, либо в состояние тайм-аута.

По истечении времени повторной передачи из состояния быстрого восстановления мы переходим в состояние избегания перегрузок, а из состояния тайм-аута в состояние медленного старта.

На основе описания переходов между фазами протокола TCP можно построить UML-диаграмму (рис. 1).

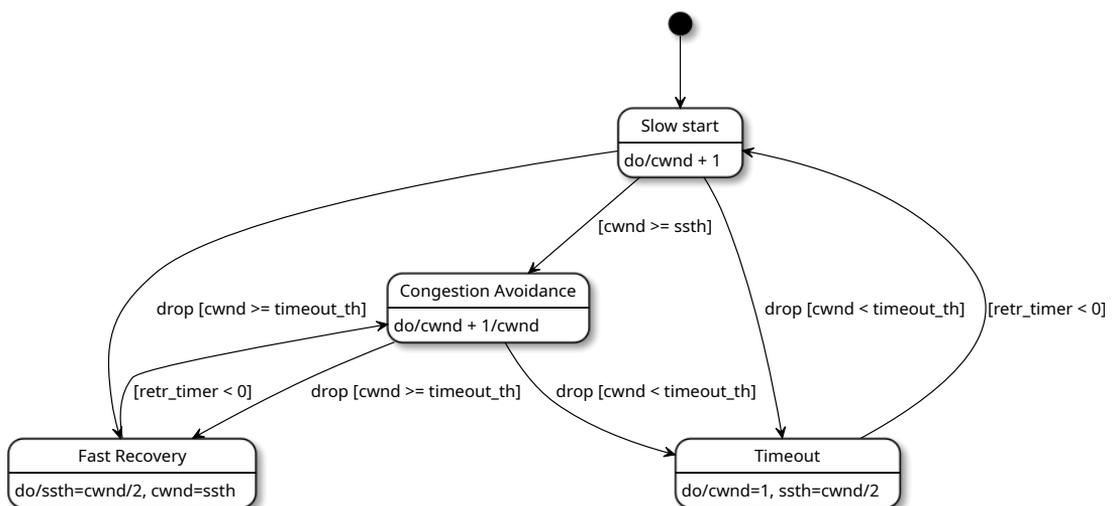


Рис. 1. Диаграмма состояний TCP

Полученную диаграмму можно преобразовать в программу на языке Modelica. Приведём фрагмент листинга (листинг 1), полностью соответствующий описанию в стандарте переходов между фазами протокола TCP.

### Листинг 1: Алгоритм перехода состояний для протокола TCP

```
algorithm
state := TCPState.slowStart;
when edge(drop_delay) and w >= timeout_th and (state ==
  TCPState.slowStart or state == TCPState.congestAvoid)
  then
    state := TCPState.fastRecov;
  elseif w >= ssth and state == TCPState.slowStart then
    state := TCPState.congestAvoid;
  elseif edge(drop_delay) and w < timeout_th and (state
    == TCPState.slowStart or state ==
    TCPState.congestAvoid) then
    state := TCPState.timeOut;
  elseif retr_timer < 0 and state == TCPState.fastRecov
    then
    state := TCPState.congestAvoid;
  elseif retr_timer < 0 and state == TCPState.timeOut
    then
    state := TCPState.slowStart;
  end when;
```

Также на языке Modelica запишем уравнения изменения размера окна передачи протокола TCP (также в соответствии с описанием, приведённом в стандарте) (листинг 2).

### Листинг 2: Изменение окна в протоколе TCP

```
// Fast Recovery
when (pre(state) == TCPState.slowStart or pre(state) ==
  TCPState.congestAvoid) and state == TCPState.fastRecov
  then
    reinit(retr_timer, o.RTT);
    reinit(ssth, w / 2);
    reinit(w, w / 2);
  end when;
// Timeout
when (pre(state) == TCPState.slowStart or pre(state) ==
  TCPState.congestAvoid) and state == TCPState.timeOut
  then
    reinit(retr_timer, RT0);
    reinit(ssth, w / 2);
    reinit(w, 1);
  end when;
```

Таким образом, мы продемонстрировали возможности языка Modelica по моделированию дискретных моделей. Фактически, дословно следуя тексту стандарта, описывающего функционирование протокола TCP, мы получили имитационную модель протокола.

#### 4. Численное моделирование системы с управлением средствами языка Modelica

В работах [23, 24] в формализме стохастических дифференциальных уравнений с пуассоновским процессом построена модель взаимодействия входящего TCP-потока и маршрутизатора, обрабатывающего поток по алгоритму типа RED. Модель имеет вид системы обыкновенных дифференциальных уравнений. Фактически на эту систему накладывается ряд ограничений, которые затрудняют исследование данной модели в рамках непрерывного подхода, однако эти ограничения вполне укладываются в гибридный подход.

Математически систему процесса передачи трафика с регулируемой алгоритмом типа RED динамической интенсивностью потока можно представить в виде следующих трёх уравнений:

$$\begin{cases} \dot{W}(t) = \frac{1}{T(t)} \vartheta(W_{\max} - W) - \frac{W(t)}{2} \frac{W(t - T(t))}{T(t - T(t))} p(t - T(t)), \\ \dot{Q}(t) = \begin{cases} \frac{N(t)W(t)}{T(t)} - C, & Q(t) > 0, \\ \max\left(\frac{N(t)W(t)}{T(t)} - C, 0\right), & Q(t) = 0, \end{cases} \\ \dot{\hat{Q}}(t) = w_q C \hat{Q}(t) + w_q C Q(t). \end{cases} \quad (4)$$

В этой системе  $W(t)$  обозначает среднее значение размера TCP окна (измеряется в пакетах), через  $Q(t)$  обозначено среднее значение длины очереди (также измеряется в пакетах), через  $\hat{Q}(t)$  — экспоненциально взвешенное скользящее среднее значение длины очереди. Кроме того, введены параметры системы:  $C$  — интенсивность обработки пакетов в очереди маршрутизатора,  $N(t)$  — число TCP-сессий,  $T(t)$  — время двойного оборота (Round Trip Time, сек). Элемент  $\vartheta(W_{\max} - W)$  является функцией Хэвисайда и ограничивает рост окна (см. листинг 3).

##### Листинг 3: Ограничение роста окна

```
function wAdd
  input Real wIn;
  input Real wmax;
  input Real T;
  output Real wOut;
algorithm
  wOut := if noEvent(wIn > wmax) then 0 else 1 / T;
end wAdd;
```

Тогда уравнение для изменения окна запишется в следующем виде (листинг 4).

##### Листинг 4: Уравнение для изменения окна

```
equation
  der(w) = wAdd(w, wmax, T) - w * delay(w, T) / (2 * delay(T, T)
  ) * delay(p, T);
```

Оператор `der` задаёт производную по времени, а оператор `delay` задаёт запаздывание. При этом заметим, что моделирование запаздывания (особенно функционального) при непрерывном подходе представляет крайне нетривиальную задачу.

Алгоритм изменения мгновенной длины очереди представлен в листинге 5.

**Листинг 5: Алгоритм изменения мгновенной длины очереди**

```
algorithm
  q1 := N * w / T - C;
  qOut := if noEvent(q + q1 > 0) then q1 else -q;
```

Тогда дифференциальное уравнение для мгновенной длины очереди имеет крайне простой вид (листинг 6).

**Листинг 6: Уравнение для изменения мгновенной длины очереди**

```
equation
  der(q) = qAdd(pre(q), w, T, C, N);
```

Непосредственно за управление по алгоритму RED [1] отвечает функция сброса пакета  $p$ , значения которой лежат в интервале  $[0, 1]$ :

$$p = \begin{cases} 0, & 0 \leq \hat{Q} < Q_{\min}, \\ \frac{\hat{Q} - Q_{\min}}{Q_{\max} - Q_{\min}} p_{\max}, & Q_{\min} \leq \hat{Q} \leq Q_{\max}, \\ 1, & \hat{Q} > Q_{\max}. \end{cases} \quad (5)$$

Как мы видим, функция сброса — кусочная и разрывная. Всё это также затрудняет использование непрерывного подхода. В гибридном подходе функция сброса задаётся единственным атомарным оператором (см. листинг 7).

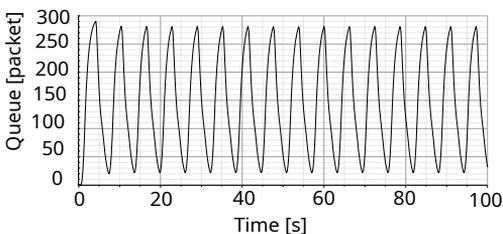
**Листинг 7: Функция сброса RED**

```
p = if ( q_avg < thmin*R ) then 0.0 elseif ( q_avg > thmax*
  R ) then 1.0 else (q_avg - thmin*R) * pmax / (thmax*R -
  thmin*R);
```

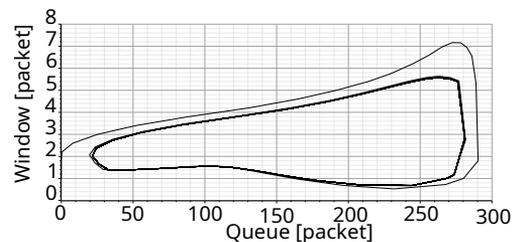
Здесь через  $R$  обозначен размер буфера, поскольку значения порогов нормированы на единицу.

Получившаяся программа компактна и имеет крайне простой вид. В чисто непрерывном подходе результирующая программа была излишне громоздкой.

Результаты вычислений можно представить в виде графиков. Как пример, приведём график поведения мгновенной длины очереди (рис. 2) и соответствующий фазовый портрет (в координатах  $w$  и  $q$ ) (рис. 3)



**Рис. 2. График мгновенной длины очереди**



**Рис. 3. Фазовый портрет mbox**

## 5. Заключение

Авторы преследовали своей целью продемонстрировать применимость гибридного подхода для задач моделирования сетевых протоколов. Сетевые протоколы обычно описываются на основе дискретно-событийного подхода. Однако использование дискретного подхода при моделировании зачастую приводит к недостаточной масштабируемости модели. В то же время применение непрерывного подхода приводит к излишнему усложнению модели. Причём большое количество ограничений при непрерывном подходе не входят непосредственно в математическое описание модели, а присутствует в виде дополнительного семантического материала.

Кроме того, гибридный подход может применяться как при аналитическом, так и при имитационном моделировании. Впрочем, как можно видеть из листингов, дискретный подход в рамках гибридного подхода носит всё-таки подчинённый характер. Поэтому и код для имитационной модели (где превалирует дискретное описание) выглядит более громоздким, чем код для вычислительной модели (где превалирует непрерывный подход, а дискретный подход носит вспомогательный характер).

Таким образом, рассмотренный в статье гибридный (непрерывно-дискретный) подход к моделированию представляется нам на данном этапе наиболее адекватным для решения задач моделирования сетевых протоколов. К сожалению, данный подход недостаточно активно применяется исследователями, хотя и реализован в ряде систем компьютерного моделирования.

## Благодарности

Публикация подготовлена при поддержке Программы РУДН «5-100» и при финансовой поддержке РФФИ в рамках научного проекта № 16-07-00556.

## Литература

1. *Floyd S., Jacobson V.* Random Early Detection Gateways for Congestion Avoidance // IEEE/ACM Transactions on Networking. — 1993. — Vol. 1, No 4. — Pp. 397–413. — DOI: 10.1109/90.251892.
2. *Altman E., Jiménez T.* NS Simulator for Beginners // Synthesis Lectures on Communication Networks. — 2012. — Vol. 5, No 1. — Pp. 1–184. — DOI: 10.2200/S00397ED1V01Y201112CNT010.
3. *Issariyakul T., Hossain E.* Introduction to Network Simulator NS2. — Boston, MA: Springer US, 2012. — P. 510. — DOI: 10.1007/978-1-4614-1406-3.
4. The Method of Constructing Models of Peer to Peer Protocols / A. V. Demidova, A. V. Korolkova, D. S. Kulyabov, L. A. Sevastyanov // 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). — IEEE Computer Society, 2015. — Pp. 557–562. — DOI: 10.1109/ICUMT.2014.7002162.
5. One-Step Stochastic Processes Simulation Software Package / E. G. Eferina, A. V. Korolkova, M. N. Gevorkyan et al. // Bulletin of Peoples' Friendship University of Russia. Series "Mathematics. Information Sciences. Physics". — 2014. — No 3. — Pp. 46–59.
6. Congestion Control Scheme Performance Analysis Based on Nonlinear RED / C.-W. Feng, L.-F. Huang, C. Xu, Y.-C. Chang // IEEE Systems Journal. — 2015. — Pp. 1–8. — DOI: 10.1109/JSYST.2014.2375314.
7. *Lautenschlaeger W., Francini A.* Global Synchronization Protection for Bandwidth Sharing TCP Flows in High-Speed Links // Proc. 16-th International Conference on High Performance Switching and Routing, IEEE HPSR 2015. — Budapest, Hungary: 2015.
8. *Karmeshu, Patel S., Bhatnagar S.* Adaptive Mean Queue Size and Its Rate of Change: Queue Management with Random Dropping. — 2016. — Pp. 1–17.

9. *Paxson V., Floyd S.* Why We Don't Know How to Simulate the Internet // Proceedings of the 29th conference on Winter simulation - WSC '97. — New York, USA: ACM Press, 1997. — Pp. 1037–1044. — DOI: 10.1145/268437.268737.
10. *Paxson V., Floyd S.* Wide Area Traffic: the Failure of Poisson Modeling // IEEE/ACM Transactions on Networking. — 1995. — Vol. 3, No 3. — Pp. 226–244. — DOI: 10.1109/90.392383.
11. On the Self-Similar Nature of Ethernet Traffic (Extended Version) / W. E. Leland, M. S. Taqqu, W. Willinger, D. V. Wilson // IEEE/ACM Transactions on Networking. — 1994. — Vol. 2, No 1. — Pp. 1–15. — DOI: 10.1109/90.282603.
12. *Maler O.* Hybrid Systems and Real-World Computations // Workshop on Theory of Hybrid Systems. — Lyndby, Denmark: Springer-Verlag, 1992.
13. *Maler O.* Control from Computer Science // Annual Reviews in Control. — 2002. — Vol. 26, No 2. — Pp. 175–187. — DOI: 10.1016/S1367-5788(02)00030-5.
14. Hybrid Modeling of Communication Networks Using Modelica / D. Färnqvist, K. Strandemar, K. H. Johansson, J. P. Hespanha // The 2nd International Modelica Conference. — 2002. — Pp. 209–213.
15. Hybrid Modeling of TCP Congestion Control / J. P. Hespanha, S. Bohacek, K. Obraczka, J. Lee // Lncs. — 2001. — 2034. — Pp. 291–304. — DOI: 10.1007/3-540-45351-2.25.
16. *Bohacek S., Lee J.* Analysis of a TCP Hybrid Model // Proc. of the 39th Annual Allerton Conference on Communication, Control, and Computing. — 2001. — Pp. 1–10.
17. Hybrid Simulation Of Active Traffic Management / A. V. Korolkova, T. R. Velieva, P. A. Abaev et al. // Proceedings 30th European Conference on Modelling and Simulation. — 2016. — Pp. 685–691. — DOI: 10.7148/2016-0685.
18. *Fritzson P.* Principles of Object-Oriented Modeling and Simulation with Modelica 2.1. — Wiley-IEEE Press, 2003. — P. 939.
19. *Fritzson P.* Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica. — Hoboken, NJ, USA: John Wiley & Sons, Inc., 2011. — DOI: 10.1002/9781118094259.
20. *Fall K., Floyd S.* Simulation-Based Comparisons of Tahoe, Reno and SACK TCP // ACM SIGCOMM Computer Communication Review. — 1996. — Vol. 26, No 3. — Pp. 5–21. — DOI: 10.1145/235160.235162.
21. *Allman M., Paxson V., Blanton E.* TCP Congestion Control. — 2009. — DOI: 10.17487/rfc5681.
22. *Brockett R.* Stochastic Analysis for Fluid Queueing Systems // Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304). — Vol. 3. — IEEE, 1999. — Pp. 3077–3082. — DOI: 10.1109/CDC.1999.831407.
23. *Misra V., Gong W.-B., Towsley D.* Stochastic Differential Equation Modeling and Analysis of TCP-WindowSize Behavior // Proceedings of PERFORMANCE. — 1999. — Vol. 99.
24. *Misra V., Gong W.-B., Towsley D.* Fluid-Based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED // ACM SIGCOMM Computer Communication Review. — 2000. — Vol. 30, No 4. — Pp. 151–160. — DOI: 10.1145/347057.347421.
25. *Velieva T. R., Korolkova A. V., Kulyabov D. S.* Designing Installations for Verification of the Model of Active Queue Management Discipline RED in the GNS3 // 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). — IEEE Computer Society, 2015. — Pp. 570–577. — DOI: 10.1109/ICUMT.2014.7002164.
26. Modeling TCP Throughput / J. Padhye, V. Firoiu, D. Towsley, J. Kurose // ACM SIGCOMM Computer Communication Review. — 1998. — Vol. 28, No 4. — Pp. 303–314. — DOI: 10.1145/285243.285291.

UDC 004.021:519.6

DOI: 10.22363/2312-9735-2018-26-4-371-382

## Computational and Simulation Models of the Control System on Modelica

Anne Marie Yu. Apreutesey\*, Anna V. Zavozina\*, Anna V. Korolkova\*,  
Dmitry S. Kulyabov\*<sup>†</sup>

\* *Department of Applied Probability and Informatics  
Peoples' Friendship University of Russia (RUDN University)  
6, Miklukho-Maklaya str., Moscow, 117198, Russian Federation*

<sup>†</sup> *Laboratory of Information Technologies  
Joint Institute for Nuclear Research  
6 Joliot-Curie, Dubna, Moscow region, 141980, Russian Federation*

When modeling network protocols, the choice of a model approach and a software implementation tool is a problem. The specificity of this subject area is that for the description of protocols usually the discrete-event approach is used. However, the discrete model approach has several disadvantages. It is poorly scalable, not well suited for describing dynamic systems. As an alternative to the discrete approach, a continuous approach is usually considered. But when modeling discrete events, continuous description becomes unnecessarily complicated and heavy. Events take the form of some restrictions on the continuous system, which are often not explicitly included in the continuous model, but have the form of additional semantic descriptions. The authors propose to use a hybrid (continuous-discrete) approach when modeling such systems. In the framework of the hybrid approach, the discrete system is recorded in a continuous form, and the events take the form of discrete transitions inherent in the approach. In addition, if it is based on the description of events, a simulation model can be obtained on the basis of a hybrid approach.

This paper demonstrates the use of a hybrid approach to describe systems with control by the example of the interaction of the TCP protocol and the RED algorithm. The simplicity of creating both computational and simulation models of the system is demonstrated. The Modelica language is used as the implementation language.

**Key words and phrases:** active queue management, simulation, hybrid modeling, Modelica, Random Early Detection (RED)

## References

1. S. Floyd, V. Jacobson, Random Early Detection Gateways for Congestion Avoidance, *IEEE/ACM Transactions on Networking* 1 (4) (1993) 397–413. doi:10.1109/90.251892.
2. E. Altman, T. Jiménez, NS Simulator for Beginners, *Synthesis Lectures on Communication Networks* 5 (1) (2012) 1–184. doi:10.2200/S00397ED1V01Y201112CNT010.
3. T. Issariyakul, E. Hossain, *Introduction to Network Simulator NS2*, Springer US, Boston, MA, 2012. doi:10.1007/978-1-4614-1406-3.
4. A. V. Demidova, A. V. Korolkova, D. S. Kulyabov, L. A. Sevastyanov, The Method of Constructing Models of Peer to Peer Protocols, in: 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), IEEE Computer Society, 2015, pp. 557–562. arXiv:1504.00576, doi:10.1109/ICUMT.2014.7002162.
5. E. G. Eferina, A. V. Korolkova, M. N. Gevorkyan, D. S. Kulyabov, L. A. Sevastyanov, One-Step Stochastic Processes Simulation Software Package, *Bulletin of Peoples' Friendship University of Russia. Series "Mathematics. Information Sciences. Physics"* (3) (2014) 46–59. arXiv:1503.07342.
6. C.-W. Feng, L.-F. Huang, C. Xu, Y.-C. Chang, Congestion Control Scheme Performance Analysis Based on Nonlinear RED, *IEEE Systems Journal* (2015) 1–8 doi:10.1109/JSYST.2014.2375314.

7. W. Lautenschlaeger, A. Francini, Global Synchronization Protection for Bandwidth Sharing TCP Flows in High-Speed Links, in: Proc. 16-th International Conference on High Performance Switching and Routing, IEEE HPSR 2015, Budapest, Hungary, 2015. arXiv:1602.05333.
8. Karmeshu, S. Patel, S. Bhatnagar, Adaptive Mean Queue Size and Its Rate of Change: Queue Management with Random Dropping (2016) 1–17arXiv:1602.02241.
9. V. Paxson, S. Floyd, Why We Don't Know How to Simulate the Internet, in: Proceedings of the 29th conference on Winter simulation - WSC '97, ACM Press, New York, USA, 1997, pp. 1037–1044. doi:10.1145/268437.268737.
10. V. Paxson, S. Floyd, Wide Area Traffic: the Failure of Poisson Modeling, IEEE/ACM Transactions on Networking 3 (3) (1995) 226–244. doi:10.1109/90.392383.
11. W. E. Leland, M. S. Taqqu, W. Willinger, D. V. Wilson, On the Self-Similar Nature of Ethernet Traffic (Extended Version), IEEE/ACM Transactions on Networking 2 (1) (1994) 1–15. doi:10.1109/90.282603.
12. O. Maler, Hybrid Systems and Real-World Computations, in: Workshop on Theory of Hybrid Systems, Springer-Verlag, Lyndby, Denmark, 1992.
13. O. Maler, Control from Computer Science, Annual Reviews in Control 26 (2) (2002) 175–187. doi:10.1016/S1367-5788(02)00030-5.
14. D. Färnqvist, K. Strandemar, K. H. Johansson, J. P. Hespanha, Hybrid Modeling of Communication Networks Using Modelica, in: The 2nd International Modelica Conference, 2002, pp. 209–213.
15. J. P. Hespanha, S. Bohacek, K. Obraczka, J. Lee, Hybrid Modeling of TCP Congestion Control, in: Lncs, no. 2034, 2001, pp. 291–304. doi:10.1007/3-540-45351-2\_25.
16. S. Bohacek, J. Lee, Analysis of a TCP Hybrid Model, in: Proc. of the 39th Annual Allerton Conference on Communication, Control, and Computing, 2001, pp. 1–10.
17. A. V. Korolkova, T. R. Velieva, P. A. Abaev, L. A. Sevastianov, D. S. Kulyabov, Hybrid Simulation Of Active Traffic Management, Proceedings 30th European Conference on Modelling and Simulation (2016) 685–691doi:10.7148/2016-0685.
18. P. Fritzson, Principles of Object-Oriented Modeling and Simulation with Modelica 2.1, Wiley-IEEE Press, 2003.
19. P. Fritzson, Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2011. doi:10.1002/9781118094259.
20. K. Fall, S. Floyd, Simulation-Based Comparisons of Tahoe, Reno and SACK TCP, ACM SIGCOMM Computer Communication Review 26 (3) (1996) 5–21. doi:10.1145/235160.235162.
21. M. Allman, V. Paxson, E. Blanton, TCP Congestion Control (sep 2009). doi:10.17487/rfc5681.
22. R. Brockett, Stochastic Analysis for Fluid Queueing Systems, in: Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304), Vol. 3, IEEE, 1999, pp. 3077–3082. doi:10.1109/CDC.1999.831407.
23. V. Misra, W.-B. Gong, D. Towsley, Stochastic Differential Equation Modeling and Analysis of TCP-Window Size Behavior, Proceedings of PERFORMANCE 99.
24. V. Misra, W.-B. Gong, D. Towsley, Fluid-Based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED, ACM SIGCOMM Computer Communication Review 30 (4) (2000) 151–160. doi:10.1145/347057.347421.
25. T. R. Velieva, A. V. Korolkova, D. S. Kulyabov, Designing Installations for Verification of the Model of Active Queue Management Discipline RED in the GNS3, in: 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), IEEE Computer Society, 2015, pp. 570–577. arXiv:1504.02324, doi:10.1109/ICUMT.2014.7002164.
26. J. Padhye, V. Firoiu, D. Towsley, J. Kurose, Modeling TCP Throughput, ACM SIGCOMM Computer Communication Review 28 (4) (1998) 303–314. doi:10.1145/285243.285291.

© Апреутесей А. М. Ю., Завозина А. В., Королькова А. В., Кулябов Д. С., 2018



This work is licensed under a Creative Commons Attribution 4.0 International License

**Для цитирования:**

*Апреутесей А. Ю., Завозина А. В., Королькова А. В., Кулябов Д. С.* Вычислительная и имитационная модели системы с управлением на Modelica // Вестник Российского университета дружбы народов. Серия: Математика. Информатика. Физика. — 2018. — Т. 26, № 4. — С. 371–382. — DOI: 10.22363/2312-9735-2018-26-4-371-382.

**For citation:**

Apreutesey A. Y., Zavozina A. V., Korolkova A. V., Kulyabov D. S. Computational and Simulation Models of the Control System on Modelica, RUDN Journal of Mathematics, Information Sciences and Physics 26 (4) (2018) 371–382. DOI: 10.22363/2312-9735-2018-26-4-371-382. In Russian.

**Сведения об авторах:**

**Апреутесей Анна Мария Юрьевна** — студент кафедры прикладной информатики и теории вероятностей РУДН (тел.: +7(495) 952-02-50)

**Завозина Анна Васильевна** — студент кафедры прикладной информатики и теории вероятностей РУДН (тел.: +7(495) 952-02-50)

**Королькова Анна Владиславовна** — доцент, кандидат физико-математических наук, доцент кафедры прикладной информатики и теории вероятностей РУДН (тел.: +7(495) 952-02-50)

**Кулябов Дмитрий Сергеевич** — доцент, доктор физико-математических наук, доцент кафедры прикладной информатики и теории вероятностей РУДН (тел.: +7 (495) 952-02-50)

**Information about the authors:**

**Anne-Marie Yu. Apreutesey** — student of Department of Applied Probability and Informatics of Peoples' Friendship University of Russia (RUDN University) (phone: +7(495) 952-02-50)

**Anna V. Zavozina** — student of Department of Applied Probability and Informatics of Peoples' Friendship University of Russia (RUDN University) (phone: +7(495) 952-02-50)

**Korolkova, Anna V.** — Associate Professor, Candidate of Sciences in Physics and Mathematics, Associate Professor of Department of Applied Probability and Informatics of Peoples' Friendship University of Russia (RUDN University) (phone: +7(495) 952-02-50)

**Kulyabov, Dmitry S.** — Associate Professor, Doctor of Sciences in Physics and Mathematics, Associate Professor of Department of Applied Probability and Informatics of Peoples' Friendship University of Russia (RUDN University) (phone: +7 (495) 952-02-50)