
UDC 004.021:519.6

Modeling RED algorithm modifications in the OpenModelica

Anna-Maria Y. Apreutesey*, Anna V. Korolkova*, Dmitry S. Kulyabov*[†]

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

*† Laboratory of Information Technologies
Joint Institute for Nuclear Research
6 Joliot-Curie, Dubna, Moscow region, 141980, Russia*

Email: 1032152610@pfur.ru, korolkova-av@rudn.ru, kulyabov-ds@rudn.ru

The goal of this work is to simulate the Random Random Detection algorithm (RED) and Double Slope Random Early Detection (DSRED) and Gentle Random Early Detection (GRED) modifications in Modelica. RED and its modifications allow to control the network load by selectively discarding packets before the queue is full and Transmission Control Protocol (TCP) begins to reduce the transmission rate preventing resynchronization. This selective packet loss helps TCP find the right data rate faster and keep the queue size and latency at the appropriate level. The existence of a large number of modifications of the classical RED algorithm is associated with the problem of selecting the parameters of the algorithm (queue thresholds, maximum drop parameter, etc.) under which the system would function stably and efficiently. The Modelica language is used as the implementation language. Based on the results obtained during the simulation it is planned to conduct a comparative analysis of the three algorithms with similar initial parameters to reveal the advantages of one or another algorithm.

Key words and phrases: active queue management, simulation, hybrid modeling, Mod-
elica.

Copyright © 2019 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors.

In: K. E. Samouylov, L. A. Sevastianov, D. S. Kulyabov (eds.): Selected Papers of the IX Conference "Information and Telecommunication Technologies and Mathematical Modeling of High-Tech Systems", Moscow, Russia, 19-Apr-2019, published at <http://ceur-ws.org>

1. Introduction

The active queue management (AQM) [1–3] algorithm with the Random Early Detection (RED) control algorithm is used to monitor and prevent congestion in router queues [4–7]. The RED allows you to adjust the flow rate by selectively dropping packets before the queue is full. If the router queue is almost empty, all packets arriving at the system are accepted. As the queue is filled and as a result of exceeding a certain threshold value, the packet drop function starts to work. This causes TCP-like protocols to slow down the transmission rate and prevents re-synchronization of connection parameters. There are many modifications of RED that improve certain characteristics of this algorithm [8, 9].

As a modeling tool is proposed to use the language Modelica in the software environment OpenModelica [10–13]. Modelica is a language developed by the non-profit organization Modelica. This association is also developing a free standard library for this language. Modelica supports continuous and hybrid (continuously discrete) paradigms.

2. Simulation of the DSRED algorithm

The algorithm Double Slope Random Early Detection (DSRED) [9, 14] introduces an additional threshold value q_{mid} between the minimum q_{min} and maximum q_{max} thresholds of the queue, $q_{mid} = 0,5(q_{max} - q_{min})$.

The probability function for dropping packets in the DSRED algorithm is:

$$p(\hat{q}) = \begin{cases} 0, & \hat{q} < q_{min}, \\ \alpha(\hat{q} - q_{min}), & q_{min} \leq \hat{q} < q_{mid}, \\ 1 - \gamma + \beta(\hat{q} - q_{mid}), & q_{mid} \leq \hat{q} < q_{max}, \\ 1, & \hat{q} \geq q_{max}. \end{cases} \quad (1)$$

$$\alpha = \frac{2(1 - \gamma)}{q_{max} - q_{min}}, \quad \beta = \frac{2\gamma}{q_{max} - q_{min}}.$$

Variable α reflects the slope of the function in the first segment between threshold values q_{min} and q_{mid} . Since overload is not critical in this segment, α is chosen in such a way as to discard as few packets as possible. The coefficient β adjusts the slope of the function for the second segment between q_{mid} and q_{max} . In this segment, the probability of dropping a packet must be high enough to warn the sender of overload. Therefore, β must be greater than α . The mode selector for setting the packet drop function is the variable γ (fig. 1).

The program, written in Modelica, reflects the behavior of the packet drop function (1) and the three main system parameters [7]:

- $w(t)$ is the average TCP window size (in packets);
- $q(t)$ is the average queue size (in packets);
- $\hat{q}(t)$ is the queue size exponentially weighted moving average of the queue size.

The following is a code fragment describing the behavior of the main parameters of the system $w(t)$, $q(t)$, $\hat{q}(t)$ controlled by the RED algorithm (listing 1). The operator `der` sets the time derivative, the operator `delay` sets the delay.

Listing 1: Changing the main parameters of the RED system

```

der(w) = wAdd(w, wmax, T) - w * delay(w, T) * delay(p, T) /
  ↪ (2 * delay(T, T));
der(q) = qAdd(pre(q), w, T, C, N, R);
der(q_avg) = wq * C * (q - q_avg);
when w <= 1.0 then

```

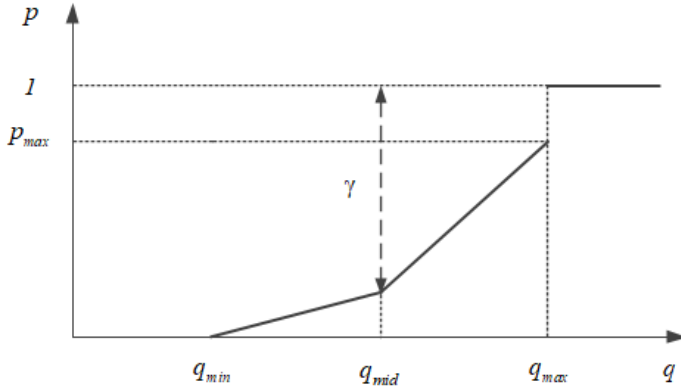


Figure 1. The packet drop function in the DSRED algorithm

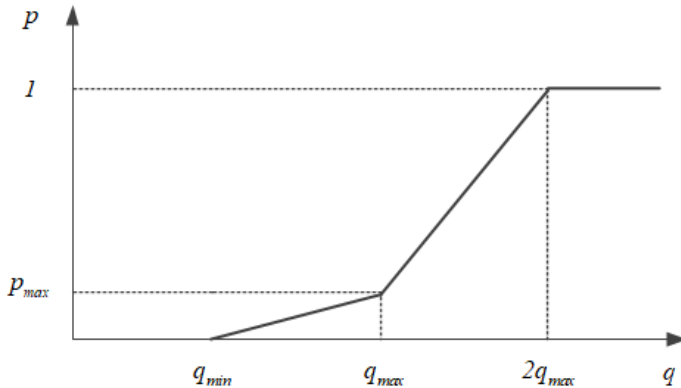


Figure 2. The packet drop function in the GRED algorithm

```

reinit(w, 1.0);
end when;
when q >= R then
  reinit(q, R);
end when;

```

The packet drop function $p(\hat{q})$, which distinguishes the DSRED algorithm from the classic RED, is described in Modelica as follows (listing 2).

Listing 2: The packet drop function in the DSRED algorithm

```

p = if q_avg < thmin * R then 0.0

```

```

elseif q_avg >= thmin * R and q_avg < thmid * R
then alfa * (q_avg - thmin * R)
elseif q_avg >= thmid * R and q_avg < thmax * R
then 1 - gamma + betta * (q_avg - thmin * R)
else 1.0;

```

The following is a listing fragment describing the growth of a TCP window (listing 3).

Listing 3: The growth of a TCP window

```

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;

```

To change the instantaneous queue length, set the appropriate function (listing 4).

Listing 4: Change of the queue length

```

function qAdd
  input Real q, w, T, C, N, R;
  output Real qOut;
protected
  Real q1, q2;
algorithm
  q1 := N * w / T - C;
  q2 := q + q1;
  qOut := if (q2 > R) then R - q elseif (q2 > 0) then q1
           ↪ else -q;
end qAdd;

```

The following variables act as initial parameters of the system:

- th_{min} is the normalized lower queue threshold;
- th_{max} is the normalized high queue threshold;
- p_{max} is the maximum probability of dropping packets.

As a result of the simulation, the graphs demonstrating the improvement in the behavior of the function of the average queue size $q(t)$ in the DSRED algorithm compared to RED with some parameters were obtained. Figure 3 shows that with the initial parameters $\gamma = 0, 8$, $th_{min} = 0, 25$, $th_{max} = 0, 6$, $p_{max} = 0, 2$, when using the DSRED algorithm, the amplitude and frequency of the oscillations of the average queue size decreased slightly, which has a positive effect on the behavior of the system. With initial parameters $th_{min} = 0, 25$, $th_{max} = 0, 6$, $p_{max} = 0, 01$, i.e. as the maximum probability of dropping packets p_{max} decreases, the amplitude of the oscillations of the queue length function in the DSRED algorithm is also slightly less than when using the classic RED, in which the function $q(t)$ reaches its maximum value (fig. 4). However, the frequency of oscillation of the average queue size function in DSRED increases, comparing with this function in RED (fig. 4). The simulation results (fig. 5) also show that with $\gamma = 0, 8$, $th_{min} = 0, 25$, $th_{max} = 0, 9$, $p_{max} = 0, 01$ parameters using the DSRED algorithm, the amplitude of fluctuations of the average queue size decreases slightly, and the frequency increases. When the lower threshold of the queue rises, i.e. with an increase in the length of the queue on which the packet drop function operates, it can be seen that the queue function $q(t)$ in the classical RED process in the oscillation process reaches its maximum value and is held at this level for some time. When using DSRED, the

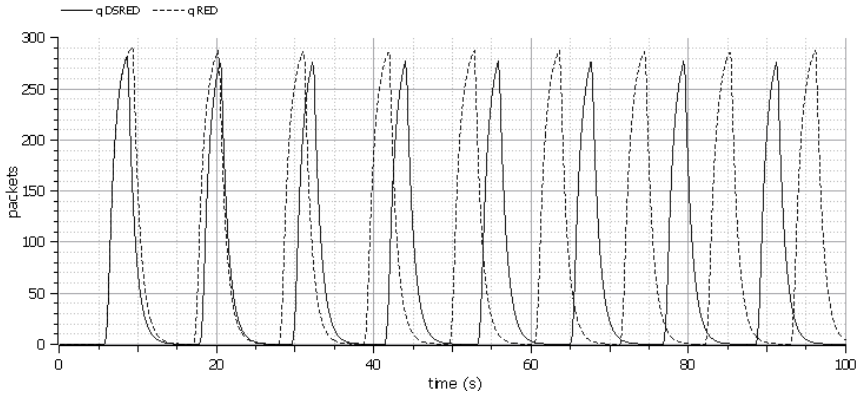


Figure 3. The average queue size function $q(t)$ in DSRED and RED algorithms, $\gamma = 0,8$, $th_{min} = 0,25$, $th_{max} = 0,6$, $p_{max} = 0,2$

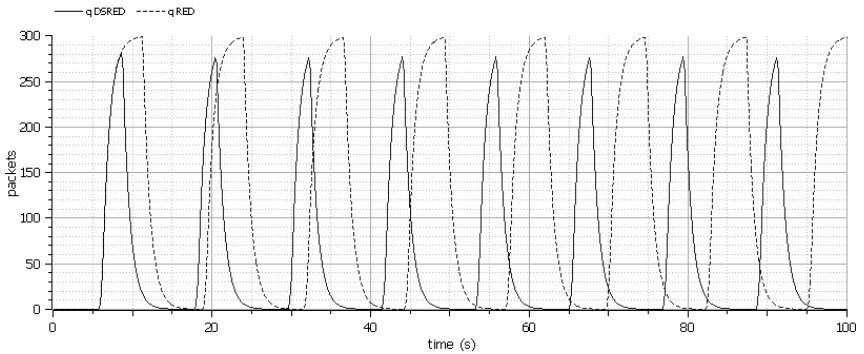


Figure 4. The average queue size function $q(t)$ in DSRED and RED algorithms, $\gamma = 0,8$, $th_{min} = 0,25$, $th_{max} = 0,6$, $p_{max} = 0,01$

function $q(t)$ does not reach the maximum possible value (fig. 5), which indicates a more stable behavior of the system parameters.

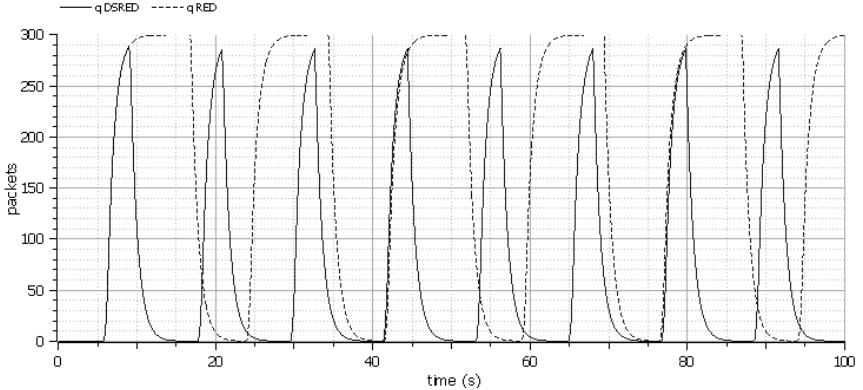


Figure 5. The average queue size function $q(t)$ in DSRED and RED algorithms, $\gamma = 0, 8$, $th_{min} = 0, 25$, $th_{max} = 0, 9$, $p_{max} = 0, 01$

3. Simulation of the GRED algorithm

In the Gentle RED algorithm (GRED) [9, 15], the packet drop function $p(\hat{q})$ is

$$p(\hat{q}) = \begin{cases} 0, & \hat{q} < q_{min}, \\ \frac{\hat{q} - q_{min}}{q_{max} - q_{min}} p_{max}, & q_{min} \leq \hat{q} < q_{max}, \\ \frac{\hat{q} - q_{max}}{q_{max}} (1 - p_{max}) + p_{max}, & q_{max} \leq \hat{q} < 2q_{max}, \\ 1, & \hat{q} \geq 2q_{max}. \end{cases} \quad (2)$$

In the GRED algorithm the RED standard drop function is in the working range from q_{min} to q_{max} . The additional segment is necessary to provide smoother behavior of the reset function outside the operating range (fig. 2).

The program fragment presented below (listing 5) is written in Modelica and describes the behavior of the packet drop function $p(\hat{q})$ (2).

Listing 5: The packet drop function in the GRED algorithm

```
p = if q_avg < thmin * R then 0.0
  elseif q_avg >= thmin * R and q_avg < thmax * R
  then ( ( q_avg - thmin * R ) * pmax ) /
  ( thmax * R - thmin * R )
  elseif q_avg >= thmax * R and q_avg < 2 * thmax * R
  then ( ( q_avg - thmax * R ) * ( 1 - pmax ) ) /
  ( thmax * R ) + pmax
  else 1.0;
```

The variables th_{min} , th_{max} , p_{max} are variable initial system parameters. When $p_{max} > 0, 9$, differences in the behavior of systems operating on RED and GRED are practically not observed. The simulation results with the parameters $th_{min} = 0, 25$,

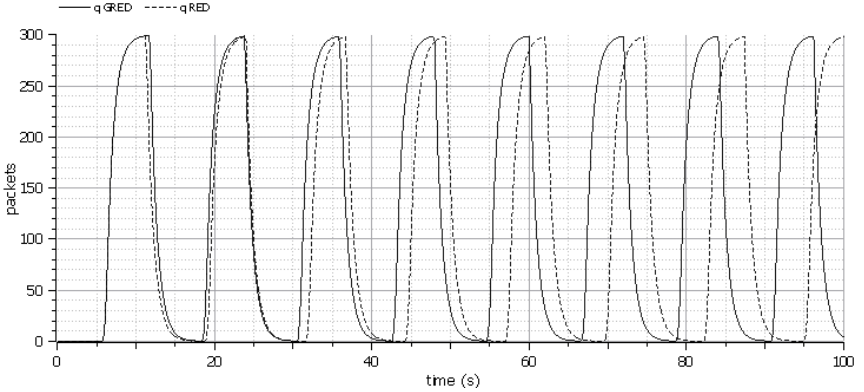


Figure 6. The average queue size function $q(t)$ in GRED and RED algorithms, $th_{min} = 0,25$, $th_{max} = 0,6$, $p_{max} = 0,01$

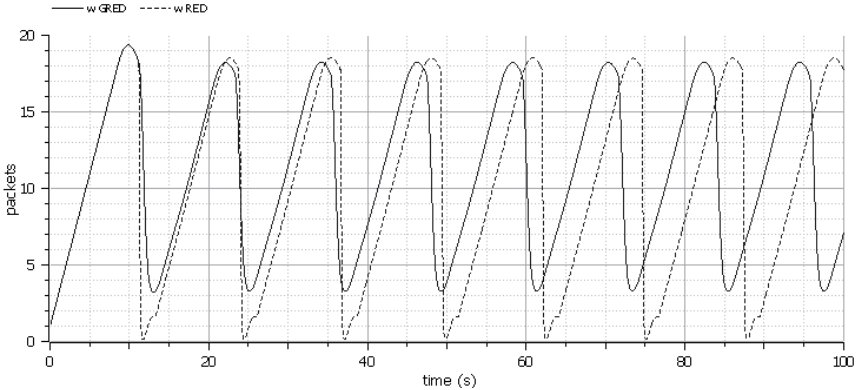


Figure 7. The TCP window size function $w(t)$ in GRED and RED algorithms, $th_{min} = 0,25$, $th_{max} = 0,6$, $p_{max} = 0,01$

$th_{max} = 0,6$, $p_{max} = 0,01$ (fig. 6) reflect that the amplitude of the average queue size with the GRED remained almost unchanged compared to the RED, and the oscillation frequency increased. Figure 7 shows the change in the average size of the TCP window. When using the GRED, the system behaves more stable, since the oscillations of the $w(t)$ in GRED do not reach zero, unlike the system according to RED. When increasing the length of the queue on which the packet drop function operates, for example with $th_{min} = 0,25$, $th_{max} = 0,9$, $p_{max} = 0,01$ parameters, it can be seen that the $q(t)$ function does not reach zero when using GRED, i.e. the amplitude of oscillation is slightly smaller compared with the same function in the RED (fig. 8). A more vivid

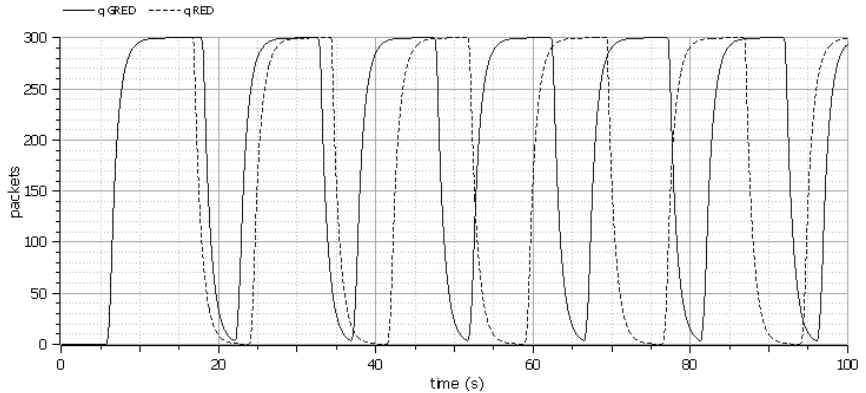


Figure 8. The average queue size function $q(t)$ in GRED and RED algorithms, $th_{min} = 0,25$, $th_{max} = 0,9$, $p_{max} = 0,01$

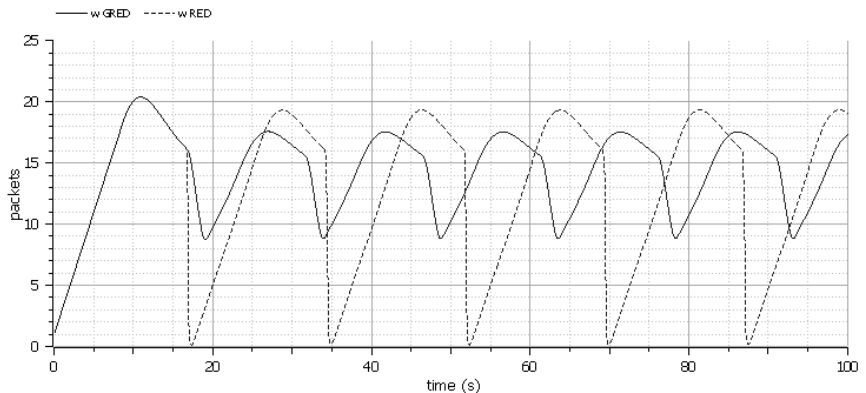


Figure 9. The TCP window size function $w(t)$ in GRED and RED algorithms, $th_{min} = 0,25$, $th_{max} = 0,9$, $p_{max} = 0,01$

example of reducing the amplitude of oscillations in a system operating on the GRED is shown in the figure 9, which shows the $w(t)$ function. The average size of a TCP window in the GRED ranges from 8,7 to 17,5 packets; in the classic RED, the $w(t)$ function ranges from 0,13 to 19,4, demonstrating the advantages of the GRED algorithm with these initial parameters.

4. Conclusions

In this paper, the principles of the operation of such RED algorithm modifications, such as DSRED and GRED, were described, mathematical models of these algorithms were given, and the systems controlled by these algorithms were simulated using the Modelica language in the OpenModelica. During the simulation, the results were obtained demonstrating that under some initial parameters the amplitude or frequency of oscillations of the average queue size can be reduced. The systems operating on DSRED and GRED, mainly showed a slight decrease in the amplitude of the oscillation parameters $q(t)$ or $w(t)$, which indicates a positive effect on the system behavior, however the oscillation frequency in most cases increased. By increasing the length of the queue on which the packet drop function operates, the DSRED algorithm made the behavior of the $q(t)$ function more stable, reducing its amplitude. In the GRED system, the behavior of the function of the average queue size changed slightly, however, the $w(t)$ function graphs showed a more stable behavior of the system than at lower maximum threshold values of the queue.

Acknowledgments

The publication has been prepared with the support of the “RUDN University Program 5-100” (Anna-Maria Y. Apreutesey, Anna V. Korolkova, calculation model development and algorithms on Modelica language). The reported study was funded by Russian Foundation for Basic Research (RFBR), project number 19-01-00645 (Dmitry S. Kulyabov, research algorithm development).

References

1. L. Chrost, A. Chydzinski, On the Evaluation of the Active Queue Management Mechanisms, 2009 First International Conference on Evolving Internet (2009) 113–118 doi:10.1109/INTERNET.2009.25.
2. R. Adams, Active Queue Management: A Survey, IEEE Communications Surveys & Tutorials 15 (3) (2013) 1425–1476. doi:10.1109/SURV.2012.082212.00018.
3. C. V. Hollot, V. Misra, D. Towsley, Wei-Bo Gong, On Designing Improved Controllers for AQM Routers Supporting TCP Flows, in: Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213), Vol. 3, IEEE, 2001, pp. 1726–1734. doi:10.1109/INFCOM.2001.916670.
4. 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.
5. V. Jacobson, Congestion Avoidance and Control, ACM SIGCOMM Computer Communication Review 18 (4) (1988) 314–329. arXiv:arXiv:1011.1669v3, doi:10.1145/52325.52356.
6. V. Misra, W.-B. Gong, D. Towsley, Stochastic Differential Equation Modeling and Analysis of TCP-Windowsize Behavior, Proceedings of PERFORMANCE 99 (1999).
7. 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.
8. A. V. Korolkova, D. S. Kulyabov, A. I. Chernoiyanov, On the Classification of RED Algorithms, Bulletin of Peoples’ Friendship University of Russia. Series “Mathematics. Information Sciences. Physics” (3) (2009) 34–46.
9. A. V. Korolkova, D. S. Kulyabov, Mathematical Model of the Dynamic Behavior of RED-Like System Parameters, Bulletin of Peoples’ Friendship University of Russia. Series “Mathematics. Information Sciences. Physics” (1) (2010) 54–64.

10. P. Fritzon, *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*, Wiley-IEEE Press, 2003.
11. P. Fritzon, *Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2011. doi:10.1002/9781118094259.
12. A. V. Korolkova, T. R. Velieva, P. O. 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.
13. T. R. Velieva, E. G. Eferina, A. V. Korolkova, D. S. Kulyabov, L. A. Sevastianov, Modelica-based TCP simulation, Journal of Physics: Conference Series 788 (100) (2017) 012036.1–7. doi:10.1088/1742-6596/788/1/012036.
14. B. Zheng, M. Atiquzzaman, DSRED: A New Queue Management Scheme for the Next Generation Internet, IEICE Transactions on Communications E89-B (3) (2006) 764–774. doi:10.1093/ietcom/e89-b.3.764.
15. G. Iannaccone, M. May, C. Diot, Aggregate traffic performance with active queue management and drop from tail, ACM SIGCOMM Computer Communication Review 31 (3) (2001) 4. doi:10.1145/505659.505661.