Interactions between AQM routers in the Internet

Pawel Mrozowski and Andrzej Chydzinski

***Abstract*—** In this paper we present a study on mutual interactions between queue management algorithms implemented in Internet routers. The aim of this research was to detect possible issues connected with deployment of the active queue management (AQM) algorithms in the Internet. For that purpose, we simulated a network with four routers and checked its performance in different configurations of queue management algorithms. In particular, three basic configurations were considered: all routers in the network use the passive queue management, only one router in the network uses the active queue management (other routers use the passive queue management), all routers in the network use the active queue management. Six different AQM algorithms were used and tested in two congestion scenarios. In every case the performance of all bottleneck links was observed as well as the average performance of the whole network. In all the simulations, the average performance of the network, measured in terms of the queueing delay, was improved by when at least one router used an AQM algorithm. Moreover, in most cases the performance of every bottleneck link was also improved. These results give strong arguments for using AQM in the Internet. However, in some experiments we observed a performance degradation of particular bottleneck links when an AQM algorithm was used, indicating that some problems may indeed arise due to the mutual interactions of AQM routers.

***Keywords*—** active queue management, Internet routers, packet queueing, performance evaluation

I. INTRODUCTION

The active queue management (AQM) algorithms (see, for instance, [1]-[24]) are designed to improve the performance of packet queueing processes in Internet routers. They are able to outperform the classic solution, which is a FIFO drop-tail queue, in many fields. In particular, they enable a significant queue size reduction, a significant queueing delay reduction, desynchronization of TCP sources, throughput maximization, inter-flow fairness optimization and other.

All new AQM solutions are usually tested using a discrete-event simulators (e. g. Ns2, Opnet Modeller etc.). Typically, a dumb-bell network topology is used for performance evaluation. In this topology there are two routers, A and B, connected by a bottleneck link. Several external nodes are connected to these routers and they send data using the bottleneck link and the routers. The dumb-bell topology is very important for practical reasons and representative enough for basic performance evaluation purposes. However, simulation studies based on this topology cannot give answers to some important questions connected with the possibility of wide deployment of the AQM algorithms over the Internet (although recommended by IETF, AQM is not widely deployed yet). In particular, the following issues have to be considered:

1) Will it be beneficial to implement an AQM algorithm on only one router in a large network, composed of many bottlenecks and routers? (Migration problem).
2) What if we implement an AQM algorithm on several routers simultaneously - can mutual interactions of several AQM routers lead to undesired performance degradation?

Naturally, the answers to these questions have a deep impact on the possible deployment of the AQM mechanisms in the Internet.

In this paper we try to answer these questions using a new set of simulation scenarios and traffic models [7] and a network topology that consists of four routers and three bottleneck links. We use six queue management algorithms and compare the network performance in three cases: (a) no AQM in the network, (b) AQM implemented on one router only, (c) all routers use the same AQM.

The paper is structured as follows. In Section II, a detailed description of the simulation scenarios is given. In particular, it includes the network topology, link parameterizations and traffic characteristic. In Section III the AQM algorithms used in the simulations are described. In Section IV, simulation results are presented and discussed. Finally, remarks concluding the paper are given in Section V.

II. SIMULATION SCENARIOS

We used in simulations the parking-lot topology with four routers A, B, C, D, three bottleneck links A-B, B-C, C-D and four network nodes N1, N2, N3, N4 connected to routers A, B, C, D, respectively (see Fig. 1). As for the links' characteristics and traffic generators, we followed the recent proposition of a standardized benchmark suite for AQM evaluation [7].

Namely, the links' bandwidths between the nodes and the routers were set to 1Gb/s. The bottleneck links between the routers were set to 100Mb/s. The propagation delays for all links were unified and set to 10ms.

The TCP sources were located in nodes N1, N2, N3 and they transmitted packets to N2, N3, N4 using the following transmission paths N1-N2, N2-N3, N3-N4 and N1-N4.

The number of TCP connections on each of the transmission paths, n, depended on the assumed congestion scenario. We considered two congestion scenarios:

(a) mild congestion with $n = 50$,
(b) heavy congestion with $n = 500$.

Therefore, the total number of TCP flows in the bottleneck links A-B, B-C, C-D was 100 in the mild congestion scenario and 1000 in the heavy congestion scenario.

90% of the TCP flows on each transmission path were using 1500 bytes long packets and the remaining 10% of flows were using 536 bytes long packets. All 536-bytes-long flows and

---

This is extended version of the paper [20] presented during Applied Computing Conference, Athens, September 27-29, 2009.

The authors are with the Silesian University of Technology, Institute of Computer Sciences. Address: Akademicka 16, 44-100 Gliwice, Poland. (corresponding author's e-mail: Andrzej.Chydzinski@polsl.pl)

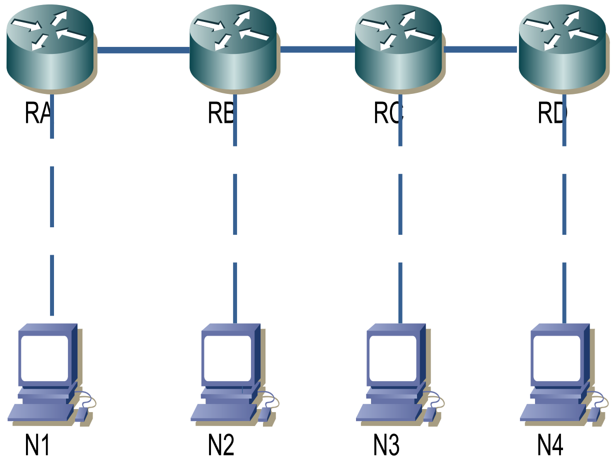


Fig. 1. The parking network topology.

75% of the 1500-bytes-long flows were long lived, FTP-like flows, transmitting large files (elephants). On the remaining 25% of the 1500-bytes-long connections a HTTP-like traffic was simulated using a generator described in [7]. Namely, a large number of small files (mice) was transmitted using each connection. Transmission start times of these small files were chosen according to a Poisson process with rate 1.25 (mild congestion) and 0.125 (heavy congestion). The file sizes were generated with Pareto distribution with the mean size of 50 kbytes and the shape parameter of 1.3.

To simulate the reverse traffic, we used UDP (CBR) flows in the same transmission paths N1-N2, N2-N3, N3-N4, N1-N4, but in the reverse direction. The number of UDP flows in each transmission path was set to $0.1n$ (i.e. 5 or 50, depending on the congestion scenario). All UDP flows used 1000 bytes long datagrams. The sending rate of the UDP flows was chosen so that the UDP traffic occupied 10% of the bandwidth of the bottleneck links (in the reverse direction).

The duration of the simulation was set to 160 seconds, although the metrics were collected from 30th to 100th second (to avoid the influence of the transient behaviour of the TCP).

The following metrics were gathered: the aggregate throughput, the average queue-length and the standard deviation of the queue length.

The simulations were performed using the Ns2 simulator (ver. 2.33). All the buffer sizes were set to 900 packets (which is an approximation of the bandwidth-delay product with the average delay of 100ms).

III. AQM ALGORITHMS

In addition to the passive queue management algorithm (FIFO drop-tail qeueue, denoted as DT), five AQM algorithms were used: RED [10], [11], PI [16], REM [1], AVQ [18] and AN-AQM [21]. Default implementations and parameterizations built in Ns2 of RED, PI, REM and AVQ were used while AN-AQM was implemented and parameterized following the description given in [21].

RED (Random Early Detection) is the most recognized AQM, proposed by Sally Floyd and Van Jacobson in 1993. Routers equipped with this AQM accept all incoming packets when the queue occupancy is below a minimum threshold. Whet the queue gets larger, the packets are dropped with probability that increases linearly with the queue length. When maximum threshold parameter is reached all the arriving packets are dropped. An important feature of the RED is the coupling of drop probability value with network congestion. RED minimizes the effect of global synchronization of TCP flows by randomly choosing and dropping incoming packets, which may belong to different flows.

REM's (Random Exponential Marking) idea is to decouple congestion measure from performance measure. REM adjusts the amount of user's data rate to the link capacity and keeps the queue size at possibly small level. During the calculation of the dropping probability of the incoming packets, special value derived from the state of the queue and the link, called ”price”, is used. The REM dropping probability function is not linear but concave exponential.

The idea of PI is taken from the classic control theory and the algorithm incorporates the well-known proportional integrator controller. The controller forces convergence of the queue size to a target value, at the same time forcing convergence of the derivative of the queue size to zero. The later goal can be achieved only when the input flow rates are matched to the link capacity and the queue size has not tendency to increase or decrease. The main advantages of PI are its good responsivenes and good ability to regulate the steady-state queue size at a target level.

The forth AQM used is our simulations is AVQ (Adaptive Virtual Queue). This AQM exploits a concept of the virtual queue, which, rather than the actual queue, is used to decide whether an incoming packet should be dropped or not. The output rate of the virtual queue is a little bit smaller than the rate of the actual link. Upon each packet arrival, both the real and the virtual queue are increased. The incoming packet is dropped when the virtual queue is full, what happens before the actual queue can gets full. Such approach enables the system to control the link utilization instead of the queue size.

Finally, the AN-AQM is a novel approach, which utilizes an adaptive neuron PID controller. It uses combined Hebbian Learning and Supervised Learning to achieve the main target - queue size stabilization.

IV. PERFORMANCE RESULTS

Now we can present simulation results. In Tabs. I-III results for the mild congestion scenario are presented, while in Tabs. IV-VI results for the heavy congestion scenario are given. In particular, in Tabs. I and IV the average queue size is presented, in mild and heavy congestion scenarios, respectively. Similarly, in Tabs. II and V the average throughput is shown, while in Tabs. III and VI the standard deviation of the queue size is shown.

The following nomenclature is used in Tabs. I-VI. For instance, ”DT-RED-DT-DT” denotes configuration with drop-tail algorithm implemented on routers A, C, D and RED algorithm implemented on router B only. ”AB queue” denotes the output queue in router A. An abbreviated form ”AN” is used to denote the AN-AQM algorithm.

As regards the queue size in the mild congestion scenario (Tab. I), we see that in all cases implementing an AQM on one router improves the average network performance and in most cases does not have a negative impact on other, non-AQM routers. On the contrary - implementing an AQM on the router B has often a positive effect on the router A, even if it has no AQM (see results for DT-RED-DT-DT, DT-REM-DT-DT etc.). An exception here is the AN-AQM algorithm, which implemented on one router can have a negative impact on other, non AQM routers (the queue size increased by 20 percent).

In Tab. II we can observe that implementing an AQM on a router may reduce the throughput a little bit, but this effect is not destructive and presents a good tradeoff when taking into account a significant queue size reduction shown in Tab. I. In some cases (e. g. PI, REM) the throughput reduction is negligible, while in the case of the AN-AQM algorithm the throughput is not reduced at all.

In Tab. III we can see that implementing an AQM has a positive impact on the queue size stability in terms of the standard deviation of the queue size. Again, AN-AQM is an exception here. It reduces the standard deviation of the queue size on AQM-operated link but may increase the standard deviation of the queue size on other, drop-tail operated links.

As regards the heavy congestion case, in Tab. IV we can see that implementing AQM usually improves the network performance. An exception here is the RED algorithm, which implemented on one router may have a negative influence on other, non-AQM routers (see RED-DT-DT-DT configuration). This observation is consistent with some previous works (for instance, the possibility of negative interactions of RED routers have been already reported in [3]). The AN-AQM algorithm does not have a negative impact on the average queue size now, but still can increase the variance of the queue size on the non-AQM routers (see Tab. VI).

It should be stressed that even if in some experiments the performance of a particular link deteriorated when an AQM was used, the average performance of the whole network was improved. This is further demonstrated in Figs. 2-5.

Namely, in Figs. 2 and 4 the mean queue size computed for three bottleneck links vs. the network configuration is depicted in the mild and heavy congestion scenarios, respectively. As we can see, in both scenarios the best results are obtained where AQM is used in all routers and the worst result are obtained if no AQM is used. In Figs. 3 and 5 the average throughput is depicted. Generally, a high throughput is preserved when AQM is used. The worst configuration reduces the throughput by 3% in the mild congestion scenario and by 2% in the heavy congestion scenario, which is acceptable taking into account drastic queue size and delay reduction. Moreover, for some AQMs the throughput is not reduced at all.

V. CONCLUSIONS

In this paper we presented a study on possible performance issues connected with deployment of AQM algorithms in large networks. Using a topology with four routers, three bottleneck links and recently proposed simulation scenarios we have shown that usage of AQM improves the average network performance, no matter if AQM is implemented on one router or on all routers. In fact, implementing an AQM on one router has often a positive impact on other, non-AQM routers. However, we observed also that in some specific scenarios implementing an AQM on one router may lead to mild performance degradation on other routers. In our experiments this effect is not critical and the overall network performance was still improved. However, it is possible that in some other specific network configuration, implementation of an AQM on one router can lead to much worse performance degradation than presented in our paper. Special care should be given to solutions based on AN-AQM and RED algorithms, because these algorithms have shown to be more prone to negative interactions than other AQMs.

	AB queue	BC queue	CD queue
DT-DT-DT-DT	492	500	498
RED-DT-DT-DT	22	411	411
DT-RED-DT-DT	22	22	399
DT-DT-RED-DT	400	22	22
DT-DT-DT-RED	405	412	22
RED-RED-RED-RED	22	22	22
REM-DT-DT-DT	76	429	427
DT-REM-DT-DT	76	76	410
DT-DT-REM-DT	405	76	75
DT-DT-DT-REM	429	431	76
REM-REM-REM-REM	75	75	75
PI-DT-DT-DT	44	412	412
DT-PI-DT-DT	44	43	406
DT-DT-PI-DT	403	43	43
DT-DT-DT-PI	414	425	43
PI-PI-PI-PI	91	114	111
AVQ-DT-DT-DT	100	391	380
DT-AVQ-DT-DT	103	97	346
DT-DT-AVQ-DT	364	94	94
DT-DT-DT-AVQ	402	406	93
AVQ-AVQ-AVQ-AVQ	83	90	91
AN-DT-DT-DT	164	604	597
DT-AN-DT-DT	165	170	603
DT-DT-AN-DT	595	170	168
DT-DT-DT-AN	593	597	166
AN-AN-AN-AN	170	170	171

TABLE I
THE AVERAGE QUEUE SIZE (IN PACKETS) IN THE MILD CONGESTION SCENARIO.

Acknowledgements: This is extended version of the paper [20] presented during Applied Computing Conference, Athens, September 27-29, 2009. This work was supported by MNiSW under grant N N516 381134.

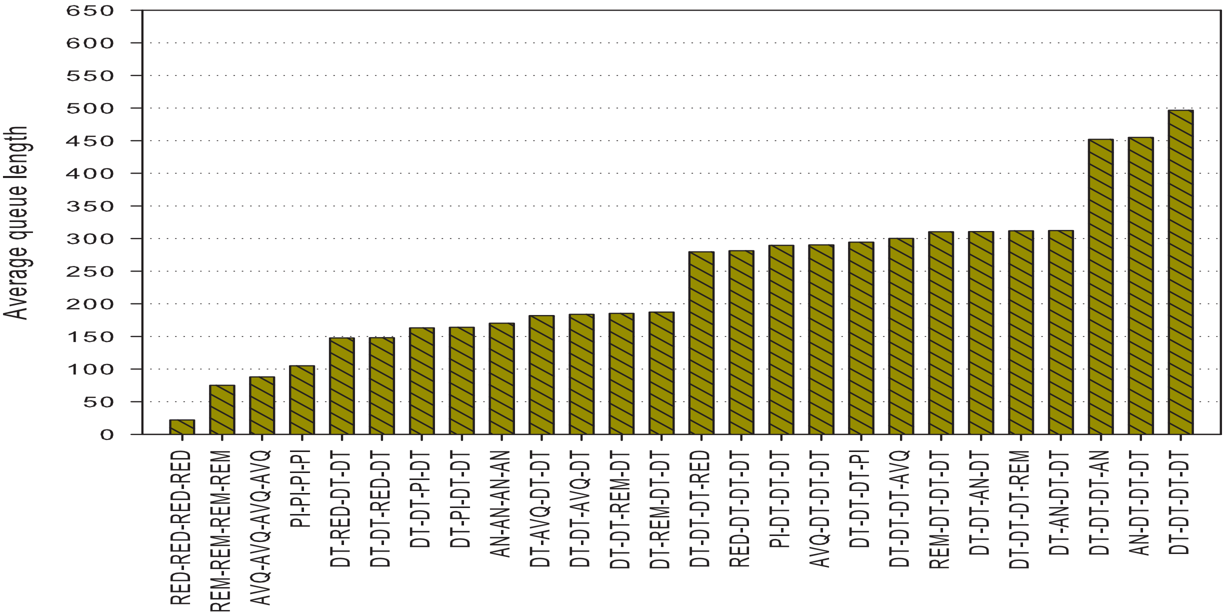


Fig. 2. The mean queue size calculated for three bottleneck links (AB, BC, CD) in the mild congestion scenario.

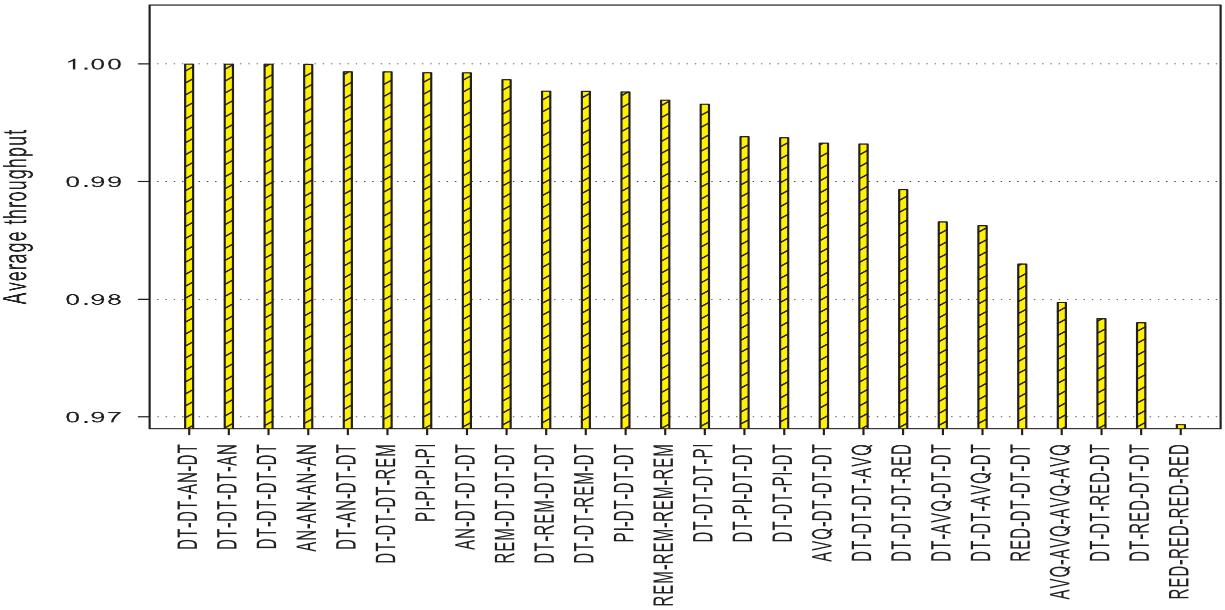


Fig. 3. The mean throughput calculated for three bottleneck links (AB, BC, CD) in the mild congestion scenario.

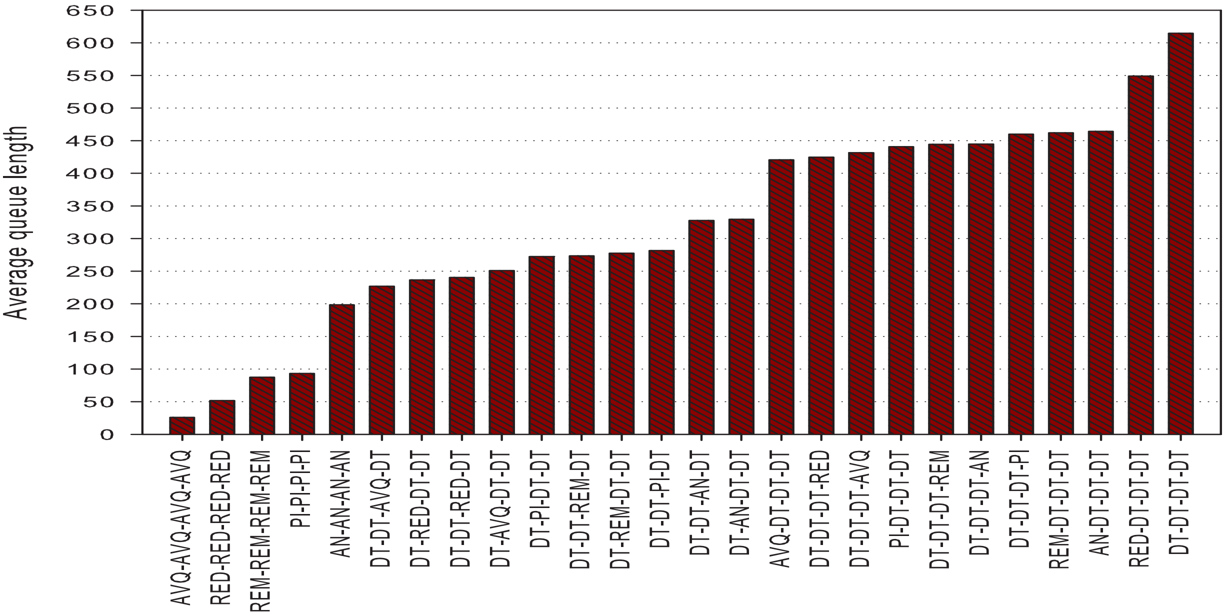


Fig. 4. The mean queue size calculated for three bottleneck links (AB, BC, CD) in the heavy congestion scenario.

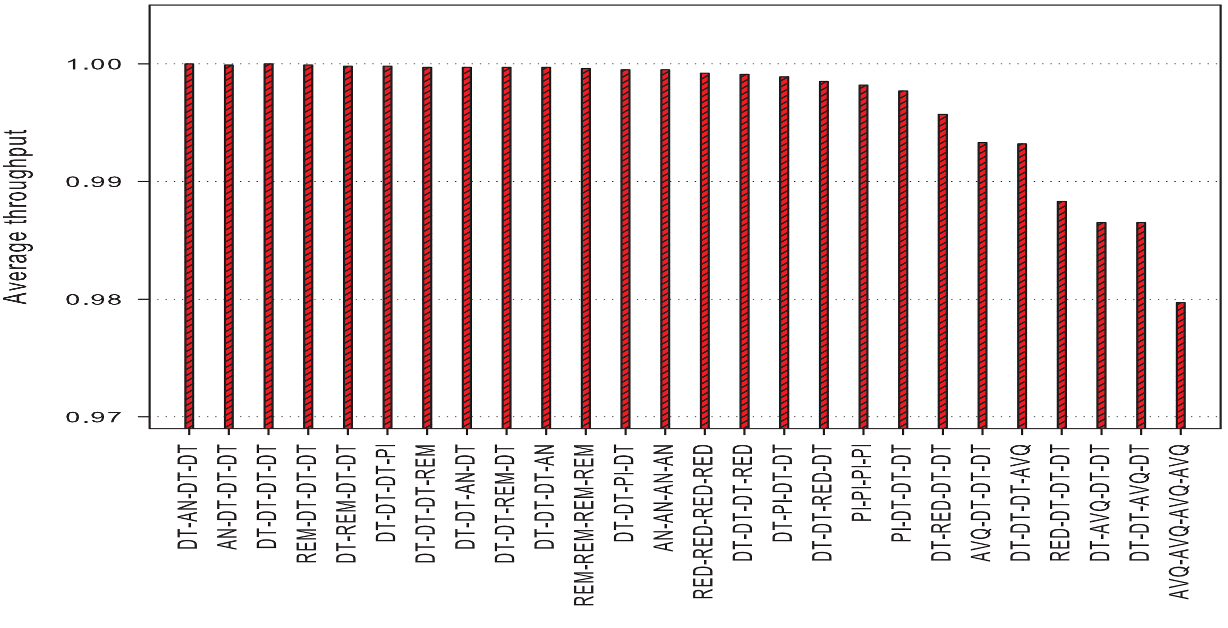


Fig. 5. The mean throughput calculated for three bottleneck links (AB, BC, CD) in the heavy congestion scenario.

	AB queue	BC queue	CD queue
DT-DT-DT-DT	100,0%	100,0%	100,0%
RED-DT-DT-DT	96,6%	100,0%	100,0%
DT-RED-DT-DT	96,6%	96,8%	100,0%
DT-DT-RED-DT	100,0%	96,6%	96,9%
DT-DT-DT-RED	100,0%	100,0%	96,8%
RED-RED-RED-RED	96,9%	96,9%	97,0%
REM-DT-DT-DT	99,6%	100,0%	100,0%
DT-REM-DT-DT	99,6%	99,7%	100,0%
DT-DT-REM-DT	100,0%	99,6%	99,7%
DT-DT-DT-REM	100,0%	100,0%	99,8%
REM-REM-REM-REM	99,8%	99,7%	99,6%
PI-DT-DT-DT	99,3%	100,0%	100,0%
DT-PI-DT-DT	99,0%	99,1%	100,0%
DT-DT-PI-DT	100,0%	99,0%	99,1%
DT-DT-DT-PI	100,0%	100,0%	99,0%
PI-PI-PI-PI	100,0%	99,9%	99,9%
AVQ-DT-DT-DT	98,0%	100,0%	100,0%
DT-AVQ-DT-DT	97,9%	98,0%	100,0%
DT-DT-AVQ-DT	100,0%	98,0%	98,0%
DT-DT-DT-AVQ	100,0%	100,0%	98,0%
AVQ-AVQ-AVQ-AVQ	98,0%	98,0%	98,0%
AN-DT-DT-DT	99,8%	100,0%	100,0%
DT-AN-DT-DT	99,8%	100,0%	100,0%
DT-DT-AN-DT	100,0%	100,0%	100,0%
DT-DT-DT-AN	100,0%	100,0%	100,0%
AN-AN-AN-AN	100,0%	100,0%	100,0%

TABLE II

THE AVERAGE THROUGHPUT IN THE MILD CONGESTION SCENARIO.

	AB queue	BC queue	CD queue
DT-DT-DT-DT	259	258	257
RED-DT-DT-DT	27	212	212
DT-RED-DT-DT	27	27	206
DT-DT-RED-DT	212	28	26
DT-DT-DT-RED	215	213	27
RED-RED-RED-RED	26	26	26
REM-DT-DT-DT	59	221	220
DT-REM-DT-DT	61	62	211
DT-DT-REM-DT	214	58	61
DT-DT-DT-REM	227	222	61
REM-REM-REM-REM	58	58	62
PI-DT-DT-DT	41	213	212
DT-PI-DT-DT	41	40	209
DT-DT-PI-DT	212	41	40
DT-DT-DT-PI	219	219	41
PI-PI-PI-PI	81	121	119
AVQ-DT-DT-DT	102	203	197
DT-AVQ-DT-DT	108	102	180
DT-DT-AVQ-DT	194	99	97
DT-DT-DT-AVQ	215	211	94
AVQ-AVQ-AVQ-AVQ	86	95	94
AN-DT-DT-DT	100	327	325
DT-AN-DT-DT	99	100	327
DT-DT-AN-DT	324	99	100
DT-DT-DT-AN	326	324	100
AN-AN-AN-AN	100	100	100

TABLE III

THE STANDARD DEVIATION OF THE QUEUE SIZE (IN PACKETS) IN THE MILD CONGESTION SCENARIO.

REFERENCES

[1] Athuraliya, S.; Low, S. H.; Li, V. H.; Qinghe Yin. REM: active queue management, IEEE Network. On page(s): 48-53, Volume: 15, Issue: 3, May 2001.

[2] Aweya J. I; Ouellette M.; Montuno D.Y. Multi-level active queue management with dynamic thresholds. Computer Communications. Volume 25, Number 8, pp. 756–771, 2002.

[3] Bauso, D.; Giarre, L. and Neglia, G. AQM stability in multiple bottleneck networks. Proc. IEEE ICC, On page(s): 2267–2271 Vol. 4, 2004.

[4] Bohacek, S.; Shah, K.; Arce, G.R.; Davis, M. Signal processing challenges in active queue management. IEEE Signal Processing Magazine, Volume. 21, Iss. 5, Pages: 69–79, Sept. 2004.

[5] Chang, X.; Muppala, J. K. A stable queue-based adaptive controller for improving AQM performance, Computer Networks: The International Journal of Computer and Telecommunications Networking, v.50 n.13, p.2204–2224, 2006.

[6] Chatranona, G., Labradorb, M.A. and Banerjee, S., A survey of TCP-friendly router-based AQM schemes. Computer Communications. v27. pp. 1424–1440, 2004.

[7] Chrost, L. and Chydzinski, A. On the Evaluation of the Active Queue Management Mechanisms. Proc. of International Conference on Evolving Internet (INTERNET'09), pp. 113-118, Cannes, Aug. 2009.

[8] Chu, H. Y.; Tsai, K. H.; Chang, W. J. Fuzzy control of active queue management routers for transmission control protocol networks via time-delay affine Takagi-Sugeno fuzzy models. International Journal of Innovative Computing, Information and Control. v4. 291-312, 2008.

[9] Fatta; G.; Hoffmann, F.; Re, G. L.; Urso, A. A genetic algorithm for the design of a fuzzy controller for active queue management, IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews. On page(s): 313-324, Volume: 33, Issue: 3, Aug. 2003.

[10] Floyd, S.; Jacobson, V. Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking, Volume 1, Issue 4, Page(s): 397 – 413, 1993.

[11] Floyd, S.; Gummadi, R. and Shenker, S. Adaptive RED: An algorithm for increasing the robustness of RED. Tech. Rep.: ACIRI, 2001.

[12] Feng, W.; Shin, K. G.; Kandlur, D. D.; Saha, D. The BLUE active queue management algorithms. IEEE/ACM Transactions on Networking. Page(s): 513 – 528, Volume: 10, Issue: 4, Aug 2002.

[13] Hariri, B.; Sadati, N. NN-RED: An AQM mechanism based on neural networks. Electronics Letters. Volume: 43, Issue: 19, On page(s): 1053–1055, 2007.

[14] Hayes, M. J.; Alavi, S. M. M.; Vandeven, P. Robust active queue management using a quantitative feedback theory based loop-shaping framework, in: Proc. American Control Conference, pp. 3077–3082, 2007.

[15] Heying, Z.; Liu, B. and Wenhua, D. Design of a robust active queue management algorithm based on feedback compensation, in: Proceedings of ACM/SIGCOMM 2003, pp. 277-285. 2002.

[16] Hollot, C. V.; Misra, V.; Towsley, D.; Weibo Gong. Analysis and design of controllers for AQM routers supporting TCP flows, IEEE Transactions on Automatic Control. On page(s): 945–959, Volume: 47, Issue: 6, Jun 2002.

[17] Hong, Y.; Yang, O. W. W. Adaptive AQM controllers for IP routers with a heuristic monitor on TCP flows: Research Articles, International Journal of Communication Systems, v.19 n.1, pp.17–38, 2006.

[18] Kunniyur, S. S.; Srikant, R. An adaptive virtual queue (AVQ) algorithm for active queue management. IEEE/ACM Transactions on Networking.

	AB queue	BC queue	CD queue
DT-DT-DT-DT	631	609	605
RED-DT-DT-DT	64	791	793
DT-RED-DT-DT	24	48	638
DT-DT-RED-DT	641	33	48
DT-DT-DT-RED	620	622	33
RED-RED-RED-RED	53	52	51
REM-DT-DT-DT	88	652	647
DT-REM-DT-DT	88	88	657
DT-DT-REM-DT	646	88	88
DT-DT-DT-REM	623	622	88
REM-REM-REM-REM	88	87	87
PI-DT-DT-DT	25	622	615
DT-PI-DT-DT	123	26	605
DT-DT-PI-DT	613	43	25
DT-DT-DT-PI	624	627	44
PI-PI-PI-PI	26	27	25
AVQ-DT-DT-DT	25	622	615
DT-AVQ-DT-DT	123	26	605
DT-DT-AVQ-DT	613	43	25
DT-DT-DT-AVQ	624	627	44
AVQ-AVQ-AVQ-AVQ	26	27	25
AN-DT-DT-DT	180	615	598
DT-AN-DT-DT	190	203	597
DT-DT-AN-DT	581	204	198
DT-DT-DT-AN	555	581	199
AN-AN-AN-AN	188	204	203

TABLE IV

THE AVERAGE QUEUE SIZE (IN PACKETS) IN THE HEAVY CONGESTION SCENARIO.

	AB queue	BC queue	CD queue
DT-DT-DT-DT	100,0%	100,0%	100,0%
RED-DT-DT-DT	98,2%	100,0%	98,3%
DT-RED-DT-DT	98,9%	99,8%	100,0%
DT-DT-RED-DT	100,0%	99,7%	99,9%
DT-DT-DT-RED	100,0%	100,0%	99,7%
RED-RED-RED-RED	99,9%	99,9%	99,9%
REM-DT-DT-DT	100,0%	100,0%	100,0%
DT-REM-DT-DT	100,0%	100,0%	100,0%
DT-DT-REM-DT	100,0%	99,9%	100,0%
DT-DT-DT-REM	100,0%	100,0%	99,9%
REM-REM-REM-REM	100,0%	100,0%	99,9%
PI-DT-DT-DT	99,3%	100,0%	100,0%
DT-PI-DT-DT	99,7%	100,0%	100,0%
DT-DT-PI-DT	100,0%	99,9%	99,9%
DT-DT-DT-PI	100,0%	100,0%	99,9%
PI-PI-PI-PI	99,7%	99,9%	99,9%
AVQ-DT-DT-DT	98,0%	100,0%	100,0%
DT-AVQ-DT-DT	98,0%	98,0%	100,0%
DT-DT-AVQ-DT	100,0%	98,0%	98,0%
DT-DT-DT-AVQ	100,0%	100,0%	98,0%
AVQ-AVQ-AVQ-AVQ	98,0%	98,0%	98,0%
AN-DT-DT-DT	99,8%	100,0%	100,0%
DT-AN-DT-DT	99,8%	100,0%	100,0%
DT-DT-AN-DT	100,0%	99,9%	100,0%
DT-DT-DT-AN	100,0%	100,0%	99,9%
AN-AN-AN-AN	100,0%	100,0%	99,8%

TABLE V

THE AVERAGE THROUGHPUT IN THE HEAVY CONGESTION SCENARIO.

Page(s): 286–299, Volume: 12, Issue: 2, April 2004.

[19] Lakshmikantha, A.; Beck, C. L.; Srikant, R. Robustness of real and virtual queue-based active queue management schemes, IEEE/ACM Transactions on Networking. On page(s): 81– 93, Volume: 13, Issue: 1, Feb. 2005.

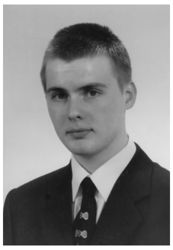
[20] Mrozowski, P.; Chydzinski, A. On the Deployment of AQM Algorithms in the Internet, Proc. of Applied Computing Conference, pp. 276–281, Athens, Sep. 2009.

[21] Sun, J. and Zukerman, M. An Adaptive Neuron AQM for a Stable Internet, Proc. Networking'07, LNCS 4479, 2007.

[22] Wang, C.; Liu, J.; Li, B. Sohraby, K; Hou, T. LRED: A Robust and Responsive AQM Algorithm Using Packet Loss Ratio Measurement. IEEE Transactions on Parallel and Distributed Systems, v.18 n.1, pp.29–43, 2007.

[23] Wu, W.; Ren, Y.; Shan, X. A self-configuring PI controller for active queue management. In Asia-Pacific Conference on Communications (APCC), Japan, 2001.

[24] Wydrowski, B. and Zukerman, M. GREEN: an active queue management algorithm for a self managed Internet, in: Proceeding of IEEE International Conference on Communications ICC2002, vol. 4, pp. 2368-2372, April, 2002.



Pawel Mrozowski received the bachelor of science degree and MSc from the University in Bielsko-Biala and the Jagiellonian University of Krakow, respectively. Currently he is a PhD student at the Silesian University of Technology. His main research interests include active queue management and network simulations. He is interested also in other areas, e. g. video compression, 3D visualizations and design, implementation and management of the enterprise hardware/software solutions. He is working also as IT specialist in hi-tech manufacturing plant, supporting team of CAD/CAM engineers.

Andrzej Chydzinski received his MS (in applied mathematics), PhD (in computer science) and habilitation (in computer science) degrees from the Silesian University of Technology, Gliwice, Poland, in 1997, 2002 and 2008, respectively. He is currently a professor in the Institute of Computer Sciences of this university. His academic and professional interests include mathematical modeling and simulation of computer networks and queueing systems. Dr Chydzinski has an established record of publications with two books and about sixty journal and conference papers.

	AB queue	BC queue	CD queue
DT-DT-DT-DT	259	258	257
RED-DT-DT-DT	36	147	150
DT-RED-DT-DT	22	35	337
DT-DT-RED-DT	346	26	36
DT-DT-DT-RED	337	333	26
RED-RED-RED-RED	38	38	37
REM-DT-DT-DT	59	341	337
DT-REM-DT-DT	58	66	341
DT-DT-REM-DT	345	66	67
DT-DT-DT-REM	339	334	66
REM-REM-REM-REM	59	66	66
PI-DT-DT-DT	50	335	334
DT-PI-DT-DT	56	88	338
DT-DT-PI-DT	343	91	92
DT-DT-DT-PI	345	335	86
PI-PI-PI-PI	56	95	95
AVQ-DT-DT-DT	19	333	331
DT-AVQ-DT-DT	111	18	331
DT-DT-AVQ-DT	339	46	18
DT-DT-DT-AVQ	339	335	47
AVQ-AVQ-AVQ-AVQ	18	20	19
AN-DT-DT-DT	103	336	333
DT-AN-DT-DT	104	106	332
DT-DT-AN-DT	328	106	105
DT-DT-DT-AN	331	331	105
AN-AN-AN-AN	103	106	106

TABLE VI
THE STANDARD DEVIATION OF THE QUEUE SIZE (IN PACKETS) IN THE HEAVY CONGESTION SCENARIO.