

Politechnika Śląska  
Wydział Automatyki, Elektroniki  
i Informatyki

mgr inż. Dariusz Marek

Aktywne zarządzanie pakietami w węźle  
komunikacyjnym z wykorzystaniem  
kontrolera PID niecałkowitego rzędu

Rozprawa doktorska napisana pod kierunkiem  
dr hab. inż. Adama Domańskiego, Prof. PŚ

Gliwice 2022



# Spis treści

|   |           |
|---|-----------|
| <b>Lista publikacji zawartych w rozprawie doktorskiej</b>   | <b>4</b>  |
| <b>Pozostałe publikacje</b>   | <b>5</b>  |
| <b>Dorobek naukowy</b>  | <b>8</b>  |
| <b>Streszczenie</b>   | <b>10</b> |
| <b>Abstract in English</b>  | <b>12</b> |
| <b>1 Wstęp</b>  | <b>14</b> |
| <b>2 Rachunek różniczkowy i różnicowy ułamkowego rzędu</b>  | <b>17</b> |
| 2.1 Funkcja $\Gamma(x)$ . . . . .   | 18        |
| 2.2 Funkcja Mittag-Lefflera . . . . .   | 19        |
| 2.3 Układy ciągłe . . . . .   | 21        |
| 2.3.1 Definicja Riemanna-Liouville'a . . . . .  | 21        |
| 2.3.2 Definicja Caputo . . . . .  | 23        |
| 2.3.3 Definicja Grünwalda-Letnikowa . . . . .   | 24        |
| 2.3.4 Operatory różniczkowania ułamkowego rzędu . . . . .   | 25        |
| 2.3.5 Różniczkowanie niecałkowitego rzędu . . . . .   | 25        |
| 2.4 Układy dyskretne . . . . .  | 26        |
| 2.4.1 Różnica niecałkowitego rzędu zapisana macierzowo . . . . .  | 28        |
| 2.4.2 Różnica niecałkowitego rzędu - implementacja . . . . .  | 29        |
| <b>3 Kontroler ułamkowego rzędu <math>PI^\alpha D^\beta</math> jako mechanizm aktywnego zarządzania kolejką</b> | <b>30</b> |

|           |  |            |
|-----------|--|------------|
| <b>4</b>  | <b>Analityczne metody ewaluacji mechanizmów aktywnego zarządzania kolejką</b>                                  | <b>36</b>  |
| 4.1       | Aproksymacja Fluid-Flow . . . . .  | 36         |
| 4.2       | Aproksymacja Fluid-Flow dla modelu sieci wielu stanowisk obsługi   | 38         |
| 4.3       | Aproksymacja dyfuzyjna . . . . .   | 38         |
| 4.3.1     | Model nieskończonej kolejki: G/G/1 . . . . .   | 40         |
| 4.3.2     | Model ograniczonej kolejki: G/G/1/N . . . . .  | 44         |
| 4.4       | Model TCP/AQM aproksymacji dyfuzyjnej . . . . .  | 46         |
| 4.4.1     | Czysty model aproksymacji dyfuzyjnej . . . . .   | 46         |
| 4.4.2     | Model łączący symulację zdarzeń dyskretnych z aproksymacją dyfuzyjną . . . . .                                 | 47         |
| 4.5       | Model dla wielu niezależnych strumieni TCP i UDP . . . . .   | 47         |
| 4.6       | Model sieci wielu stanowisk obsługi . . . . .  | 50         |
| <b>5</b>  | <b>Opis przedstawionych publikacji</b>   | <b>51</b>  |
| 5.1       | Publikacja 1: GPU accelerated non-integer order $PI^\alpha D^\beta$ controller used as AQM mechanism . . . . . | 51         |
| 5.2       | Publikacja 2: Combined diffusion approximation - simulation model of AQMs transient behaviour . . . . .        | 54         |
| 5.3       | Publikacja 3: Diffusion approximation model of TCP NewReno congestion control mechanism . . . . .              | 56         |
| 5.4       | Publikacja 4: Diffusion Model of a Non-Integer Order $PI^\gamma$ Controller with TCP/UDP Streams . . . . .     | 58         |
| 5.5       | Publikacja 5: Approximation Models for the Evaluation of TCP/AQM Networks . . . . .                            | 61         |
| <b>6</b>  | <b>GPU accelerated non-integer order <math>PI^\alpha D^\beta</math> controller used as AQM mechanism</b>       | <b>65</b>  |
| <b>7</b>  | <b>Combined diffusion approximation - simulation model of AQMs transient behaviour</b>                         | <b>80</b>  |
| <b>8</b>  | <b>Diffusion approximation model of TCP NewReno congestion control mechanism</b>                               | <b>90</b>  |
| <b>9</b>  | <b>Diffusion Model of a Non-Integer Order <math>PI^\gamma</math> Controller with TCP/UDP Streams</b>           | <b>99</b>  |
| <b>10</b> | <b>Approximation Models for the Evaluation of TCP/AQM Networks</b>   | <b>120</b> |

|  |            |
|--|------------|
| <b>11 Podsumowanie</b>   | <b>137</b> |
| <b>Bibliografia</b>  | <b>138</b> |
| <b>Oświadczenia współautorstwa</b>   | <b>147</b> |
| <b>Informacja o otrzymanych nagrodach w czasie realizacji prac nad przygotowaniem rozprawy doktorskiej</b> | <b>174</b> |

# Lista publikacji zawartych w rozprawie doktorskiej

1. Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Marek, D.; Szyguła, J. GPU accelerated non-integer order  $PI^\alpha D^\beta$  controller used as AQM mechanism. 25th International Conference on Computer Networks (CN 2018), Communications in Computer and Information Science, Springer International Publishing, vol. 860, s. 286-299, 2018. [https://doi.org/10.1007/978-3-319-92459-5\\_23](https://doi.org/10.1007/978-3-319-92459-5_23).
2. Marek, D.; Domański, A.; Domańska, J.; Czachórski T.; Klamka J.; Szyguła, J. Combined diffusion approximation - simulation model of AQMs transient behaviour. Computer Communications 2021, vol. 166, s. 40-48. <https://doi.org/10.1016/j.comcom.2020.11.014>.
3. Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Szyguła, J.; Marek, D. Diffusion approximation model of TCP NewReno congestion control mechanism. Conference on Modelling Methods in Computer Systems, Networks and Bioinformatics, 14-15 October 2019, Paris. Springer Nature Computer Science, vol. 1 (43), <https://doi.org/10.1007/s42979-019-0032-x>.
4. Marek, D.; Domański, A.; Domańska, J.; Szyguła, J.; Czachórski, T.; Klamka, J. Diffusion Model of a Non-Integer Order  $PI^\gamma$  Controller with TCP/UDP Streams. Entropy 2021, vol. 23, 619. <https://doi.org/10.3390/e23050619>.
5. Marek, D.; Domański, A.; Domańska, Szyguła, J.; Czachórski T.; Klamka J.; Filus, K. Approximation Models for the Evaluation of TCP/AQM Networks. Bulletin of the Polish Academy of Sciences: Technical Sciences, vol. 70(4), 2022. <https://doi.org/10.24425/bpasts.2022.141986>.

## Pozostałe publikacje

1. Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Marek, D.; Szyguła, J. The Influence of the Traffic Self-similarity on the Choice of the Non-integer Order  $PI^\alpha$  Controller Parameters. 32nd International Symposium on Computer and Information Sciences (ISCIS), Communications in Computer and Information Science, Springer International Publishing, vol. 935, s. 76-83, 2018. [https://doi.org/10.1007/978-3-030-00840-6\\_9](https://doi.org/10.1007/978-3-030-00840-6_9).
2. Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Marek, D.; Szyguła, J. AQM mechanism with the dropping packet function based on the answer of several  $PI^\alpha$  controllers. 26th International Conference on Computer Networks (CN 2019), Communications in Computer and Information Science, Springer International Publishing, vol. 1039, s. 400-412, 2019. [https://doi.org/10.1007/978-3-030-21952-9\\_29](https://doi.org/10.1007/978-3-030-21952-9_29).
3. Szyguła, J.; Domański, A.; Domańska, J.; Czachórski, T.; Marek, D.; Klamka, J. AQM mechanism with neuron tuning parameters. 12th Asian Conference on Intelligent Information and Database Systems (ACIIDS 2020). Lecture Notes in Artificial Intelligence, vol. 12034, s. 299-311. Springer, Cham. [https://doi.org/10.1007/978-3-030-42058-1\\_25](https://doi.org/10.1007/978-3-030-42058-1_25).
4. Remiorz, L.; Czajkowski, A.; Pawlak, S.; Remiorz, E.; Szyguła, J.; Drabik, G.; Marek, D.; Antemijczuk, O.; Paduch, J.; Baron, G.; Paszkuta, M. The concept of an interactive shower panel in terms of the assumptions of Industry 4.0. Contemporary problems of power engineering and environmental protection 2020.
5. Czachórski, T.; Gelenbe, E.; Kuaban, G. S.; Marek, D. Transient behaviour of a network router. 43rd International Conference on Telecommunications and Signal Processing. TSP. Milan, Italy, July 7-9 2020, <https://doi.org/10.1109/TSP49548.2020.9163477>.

6. Czachórski, T.; Gelenbe, E.; Kuaban, G. S.; Marek, D. Time dependent diffusion model for security driven software defined networks, The Second International Workshop on Stochastic Modeling and Applied Research of Technology (SMARTY), Petrozavodsk, Russian Federation, 16-20 August 2020, CEUR Workshop, vol. 2792, Proceedings, s.38-56, 2020.
7. Czachórski, T.; Gelenbe, E.; Marek, D. Software Defined Network Dynamics via Diffusions. 28th International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems (MASSCOTS), Nice, France, 17-19 November 2020. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 12527, s. 29-47, 2020. [https://doi.org/10.1007/978-3-030-68110-4\\_2](https://doi.org/10.1007/978-3-030-68110-4_2).
8. Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Szyguła, J.; Marek, D. The IoT gateway with active queue management. International Journal of Applied Mathematics and Computer Science, vol. 31 (1), s. 165-178, 2021. <https://doi.org/10.34768/amcs-2021-0012>.
9. Szyguła, J.; Domański, A.; Domańska, J.; Marek, D.; Filus, K.; Mendla, S. Supervised Learning of Neural Networks for Active Queue Management in the Internet. Sensors 2021, vol. 21, 4979. <https://doi.org/10.3390/s21154979>.
10. Czajkowski, A.; Remiorz, L.; Pawlak, S.; Remiorz, E.; Szyguła, J.; Marek, D.; Paszkuta, M.; Drabik, G.; Baron, G.; Paduch, J.; Antemijczuk, O. Global Water Crisis: Concept of a New Interactive Shower Panel Based on IoT and Cloud Computing for Rational Water Consumption. Appl. Sci. 2021, 11, 4081, <https://doi.org/10.3390/app11094081>.
11. Czachórski, T.; Gelenbe, E.; Kuaban, G.S.; Marek, D. Time-Dependent Performance of a Multi-Hop Software Defined Network. Appl. Sci. 2021, vol. 11, 2469. <https://doi.org/10.3390/app11062469>.
12. Gelenbe, E.; Nakıp, M.; Marek, D.; Czachórski, T. Diffusion Analysis Improves Scalability of IoT Networks to Mitigate the Massive Access Problem. 29th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), s. 1-8. <https://doi.org/10.1109/MASCOTS53633.2021.9614289>.



13. Gelenbe, E.; Nakıp, M.; Marek, D.; Czachórski, T. Mitigating the Massive Access Problem in the Internet of Things. 2nd International Symposium on Security in Computer and Information Sciences, EuroCybersec. Security in Computer and Information Sciences, Springer International Publishing, s. 118 - 132. [https://doi.org/10.1007/978-3-031-09357-9\\_10](https://doi.org/10.1007/978-3-031-09357-9_10)
14. Czachórski, T.; Gelenbe, E.; Kuaban, G. S.; Marek, D. Optimizing Energy Usage for an Electric Drone. 2nd International Symposium on Security in Computer and Information Sciences, EuroCybersec. Security in Computer and Information Sciences, Springer International Publishing, s. 61 - 75, 2022. [https://doi.org/10.1007/978-3-031-09357-9\\_6](https://doi.org/10.1007/978-3-031-09357-9_6).
15. Marek, D.; Szyguła, J; Domański, A.; Domańska, J.; Filus, K.; Szczygieł, M. Adaptive Hurst-Sensitive Active Queue Management. Entropy 2022, vol. 24(3), 418. <https://doi.org/10.3390/e24030418>.

# Dorobek naukowy

Dorobek naukowego:

- Liczba publikacji: 20
- Liczba patentów: 1
- h-index (Cytowania Scopus): 5
- Sumaryczny IF: 24,727
- Sumaryczny SNIP: 10,537
- Sumaryczny CiteScore: 39,9
- Sumaryczna punktacja MNIŚW: 1 225

Patenty:

1. Sposób wykrywania wewnętrznego osadu kamiennego w układach hydraulicznych oraz urządzenie do realizacji tego sposobu. Numer patentu: PL 240 385 B1, Data ogłoszenia patentu: 28.03.2022, Twórcy wynalazku: Sebastian Pawlak, Leszek Remiorz, Eryk Remiorz, Jakub Szyguła, Adrian Czajkowski, Gabriel Drabik, Dariusz Marek, Oleg Antemijczuk, Jarosław Paduch, Grzegorz Baron, Marcin Paszkuta.

Projekty badawcze:

1. Opracowanie systemowego oprogramowania oraz aplikacji mobilnej wraz z obsługą działania serwisu/systemu miejskiej wypożyczalni pojazdów hybrydowych i elektrycznych na obszarze Katowic, Okres uczestnictwa w projekcie: 01.12.2017 – 28.02.2018, Numer Projektu: NB-217/RM4/2017.

2. Prace badawczo-rozwojowe nad innowacyjnym wielofunkcyjnym urządzeniem sanitarnym, Okres uczestnictwa w projekcie: 02.01.2018 – 31.03.2021, Numer Projektu: POIR.01.01.01-00-0810/16.
3. Wykorzystanie operatorów ułamkowego rzędu do sterownia przeciążeniami sieci Internet, Okres uczestnictwa w projekcie: 20.06.2018 – 19.06.2023, Numer Projektu: 2017/27/B/ST6/00145.
4. Innowacyjna technologia tworzenia wydarzeń multimedialnych opartych o walki dronów z synergią między poziomami: wirtualnym, rozszerzonym i fizycznym, Okres uczestnictwa w projekcie: 01.06.2021 - 31.12.2021, Numer projektu: POIR.01.02.00-00-0160/20.
5. Technologia do bezpiecznego i niezawodnego dostarczania profesjonalnych przekazów kontrybucyjnych audio/wideo na żywo przy zachowaniu minimalnego możliwego opóźnienia, Okres uczestnictwa w projekcie: 01.08.2021 - 31.07.2022, Numer Projektu: POIR.01.01.01-00-1896/20-00.
6. Opracowanie technologii inteligentnego roju rekonfigurowalnych dronów i jej weryfikacja na przykładzie pokazu dronów i inspekcji farm fotowoltaicznych, Okres uczestnictwa w projekcie: 10.01.2022 - 31.07.2022, Numer projektu: POIR.04.01.04-00-0078/20-01.
7. Automated Guided Vehicles integrated with Collaborative Robots for Smart Industry Perspective, Okres rozpoczęcie prac w projekcie: 01.2022, Numer projektu: NOR/POLNOR/CoBotAGV/0027/2019 00.

# Streszczenie

Celem przeprowadzonych prac w ramach rozprawy doktorskiej jest ocena wykorzystania kontrolera  $PI^\alpha D^\beta$  niecałkowitego rzędu jako mechanizmu aktywnego zarządzania pakietami w węźle komunikacyjnym. Uzyskane wyniki służą ocenie zaproponowanych mechanizmów w porównaniu do już istniejących i stosowanych rozwiązań. W tym celu zastosowano symulator zdarzeń dyskretnych SimPy z wykorzystaniem języka Python oraz dwa modele analityczne - aproksymację Fluid-Flow oraz aproksymację dyfuzyjną. Samo obliczanie odpowiedzi z kontrolera  $PI^\alpha D^\beta$  jest dość czasochłonne i aby przyspieszyć ten proces zaproponowano implementację tego rozwiązania w strukturach GPU.

Niniejszą dysertację stanowi zbiór pięciu publikacji naukowych, które przedstawiają szczegółowy opis zaproponowanych modeli oraz przeprowadzonych eksperymentów badawczych, a także wnioski końcowe.

W rozdziale 1 nakreślono tematykę pracy oraz przedstawiono wybrane techniki badawcze. Zdefiniowano również cel rozprawy oraz tezę, dotyczącą zwiększenia przepustowości transmisji danych w sieci Internet.

Rachunek różniczkowy i różnicowy ułamkowego rzędu, związany bezpośrednio z obliczaniem odpowiedzi kontrolera  $PI^\alpha D^\beta$  został omówiony w rozdziale 2. Przedstawiono w nim również definicje i własności układów ciągłych oraz dyskretnych, co stanowi podstawę teoretyczną modelu zaprezentowanego w niniejszej rozprawie.

W rozdziale 3 zdefiniowano mechanizm aktywnego zarządzania kolejką w oparciu o kontroler  $PI^\alpha D^\beta$  niecałkowitego rzędu.

W kolejnym, rozdziale 4 opisano analityczne metody wykorzystane do ewaluacji modeli mechanizmów aktywnego zarządzania kolejką. W tym celu wykorzystano metody aproksymacji dyfuzyjnej oraz Fluid-Flow.

Publikacje naukowe, w oparciu o które została napisana niniejsza rozprawa, przedstawiono w rozdziale 5. Zawarto w nim również najważniejsze uzyskane rezultaty badawcze oraz wkład autorski.

W następnych rozdziałach 6-10 omówiono treść publikacji zawartych w rozprawie doktorskiej. Uzyskane rezultaty uzasadniają postawioną w tej pracy tezę. Wynika z niej, że oparcie mechanizmu aktywnego zarządzania kolejką (ang. Active Queue Management) o kontroler  $PI^\alpha D^\beta$  niecałkowitych rzędów pozwala na zmniejszenie średniej zajętości kolejki, a przez to na zwiększenie przepustowości transmisji danych w sieci Internet. Ponadto modele aproksymacji dyfuzyjnej umożliwiają ocenę kooperacji strumieni TCP i UDP z mechanizmami aktywnego zarządzania kolejką. W porównaniu do najczęściej stosowanej metody aproksymacji Fluid-Flow, aproksymacja dyfuzyjna umożliwia pozyskanie bardziej szczegółowych wyników związanych z zachowaniem kolejki oraz pozwala na modelowanie bardziej złożonych struktur sieci rozległej.

W ostatnim, rozdziale 11 podsumowano zakres przeprowadzonych prac, sformułowano wnioski końcowe oraz nakreślono potencjalne dalsze kierunki badawcze.

# Abstract in English

The purpose of the dissertation is to evaluate the use of the non-integer order  $PI^\alpha D^\beta$  controller as an Active Queue Management (AQM) mechanism implemented in a communication node. Obtained results allow to evaluate the effectiveness of proposed method and compare them to existent and used AQM mechanisms. The planned research experiments were performed based on simulation and analytical methods: Fluid-Flow approximation and diffusion approximation. The simulation model was created using the discrete event simulator SimPy and was written in Python language. The analytics models were written in Python and C languages. The  $PI^\alpha D^\beta$  computations are complex. To accelerate the computation the implementation in GPU structures was proposed.

This dissertation is a collection of five peer-reviewed research publications that present a detailed description of the proposed models and performed research experiments.

The chapter 1 introduces the topic of the dissertation and presents the selected research techniques. It also describes the aim and thesis of the dissertation.

The chapter 2 discusses Fractional Calculus, directly related to the computation of the response from non-integer order  $PI^\alpha D^\beta$  controller. It also presents definitions and properties of continuous and discrete systems. This chapter provides the theoretical basis for the model presented in this dissertation.

The chapter 3 defines an Active Queue Management mechanism based on the non-integer order  $PI^\alpha D^\beta$  controller.

The chapter 4 describes two analytical methods: Fluid-Flow approximation and diffusion approximation used to evaluate models of Active Queue Management mechanisms.

The chapter 5 describes the most important research results and presents the author's contributions.

The chapters 6-10 presents the publications included in the dissertation. The obtained results proves thesis stated in this work. Creating an Active Queue Ma-

nagement mechanism based on non-integer order  $PI^\alpha D^\beta$  controller it is possible to obtain an increase in the bandwidth in computer networks. Furthermore, the diffusion approximation model allows for assessment of the cooperation of TCP and UDP flows with the mechanisms of Active Queue Management. It allows for obtaining more detailed results of the queue behaviour than the classical method of Fluid-Flow approximation. It also enables the modeling of more complex Wide Area Network (WAN) structures.

The chapter 11 summarizes the scope of the work, presents final conclusions and proposes future works.

# Rozdział 1

## Wstęp

Problemy związane z zatłoczeniem są często spotykane w sieciach komunikacyjnych. Routery sieciowe przesyłają pakiety za pomocą łączy o ograniczonej przepustowości. Jeżeli liczba przychodzących pakietów przekracza maksymalną pojemność łącza, bufor w węzłach mogą się przepełnić.

Zgodnie ze statycznymi zasadami zarządzania kolejką, pakiety podlegają usunięciu dopiero po przekroczeniu maksymalnego poziomu zapelnienia kolejki w buforze [1]. Jednym ze sposobów usuwania nadmiarowych pakietów z kolejki jest podejście oparte o prawdopodobieństwo odrzucenia pakietu. Wyraża się je za pomocą funkcji zależnej od długości kolejki. Reguły aktywnego zarządzania kolejką (ang. Active Queue Management) polegają na nieustannym monitorowaniu stopnia zapelnienia bufora i wcześniejszym prewencyjnym usuwaniu pakietów z kolejki. Prawdopodobieństwo odrzucenia pakietu zależne jest od wartości średniej kroczącej zajętości kolejki, która wyliczana jest na podstawie aktualnego rozmiaru kolejki oraz średniej obliczonej w poprzednim slotcie [1]. Najstarszym i najbardziej rozpowszechnionym algorytmem tego typu jest mechanizm Random Early Detection (RED) [2]. W przypadku tego algorytmu funkcja odrzucania pakietów jest funkcją liniową. W ciągu ostatnich lat powstały prace, które oceniały skuteczność mechanizmu RED oraz postulowały wprowadzenie wielu zmian do tego algorytmu. Jednym z podstawowych problemów w tego typu rozwiązaniach jest prawidłowy dobór parametrów mechanizmów AQM. Zagadnienie to nie jest sprawą oczywistą. Omówienie badań związanych z doбором parametrów omówiono w artykule [3]. Natomiast w pracy [4] określono dokładne warunki, dla których system TCP/RED staje się stabilny pod względem średniej długości kolejki. W pracy tej zwrócono również uwagę, że wraz ze wzrostem przepustowości łącza, a także zmniejszeniem liczby sesji TCP wydajność mechanizmu RED



wyraźnie maleje. Kolejne znane modyfikacje polegają na zmianie sposobu kształtowania funkcji prawdopodobieństwa odrzucenia pakietu. Jednym z takich rozwiązań jest mechanizm Adaptive Random Early Detection (ARED) [5]. Głównym celem badań było przedstawienie rozwiązania, które dostosuje współczynniki mechanizmu, w szczególności wartość funkcji prawdopodobieństwa odrzucenia pakietu, w zależności od aktualnie występujących parametrów transmisji. W tym celu wprowadzono dwa dodatkowe współczynniki, które umożliwiają modyfikację maksymalnego prawdopodobieństwa utraty pakietów. Innym rozwiązaniem jest mechanizm Double Slope RED (DSRED) [6]. W tym przypadku funkcja prawdopodobieństwa odrzucenia pakietu oparta jest o dwie różne proste. Po przekroczeniu ustalonego progu zajętości kolejki, funkcja ta staje się o wiele bardziej nachylna, co skutkuje znacznym wzrostem liczby odrzucanych pakietów. Inny sposób kształtowania funkcji odrzucenia pakietów zaprezentowano w pracy [7], gdzie po raz pierwszy zaproponowano wykorzystanie do tego celu funkcji kwadratowej, w oparciu o którą stworzono mechanizm Non Linear RED (NLRED). Natomiast w następnych modyfikacjach funkcje te powiązane zostały już z krzywymi wielomianowymi [8].

Tematem niniejszej rozprawy doktorskiej jest zastosowanie koncepcji obliczania prawdopodobieństwa odrzucania pakietów, w oparciu o wykorzystanie kontrolera  $PI^\alpha D^\beta$  niecałkowitych rzędów. Odpowiedź kontrolera  $PI^\alpha D^\beta$  traktowana jest w tym przypadku jako funkcja prawdopodobieństwa odrzucenia pakietów. Przeprowadzone badania służą ocenie efektywności zaproponowanego rozwiązania, w porównaniu do już istniejących i już przebadanych mechanizmów. W czasie badań analizowane były takie parametry transmisji jak średnia długość kolejki, liczba odrzuconych pakietów oraz średni czas oczekiwania pakietu w buforze transmisyjnym. Szczegółnej analizie poddany został także wpływ parametrów kontrolera  $PI^\alpha D^\beta$  niecałkowitego rzędu na uzyskiwane rezultaty.

Prezentowane w tej pracy wyniki wykorzystują metody symulacyjne oraz metody analityczne. Modele symulacyjne [9] stosowane są zwłaszcza przy projektowaniu urządzeń sieciowych oraz poprawie ich wydajności i niezawodności. Umożliwiają one przeprowadzenie szczegółowej analizy wydajności sieci komputerowej. Narzędzia te są również przydatne do analizy zachowania pakietów [10].

W niniejszej rozprawie oprócz modeli symulacyjnych wykorzystano również dwie metody analityczne. Aproksymacja Fluid-Flow jest klasyczną metodą stosowaną do oceny współpracy mechanizmów aktywnego zarządzania kolejką z protokołem TCP [11]. Model ten ignoruje wpływ zależności czasowych i pozwala uzyskać tylko wartości średnie takich parametrów transmisji, jak np. zajętość ko-

lejkę [11]. W niniejszej pracy wykorzystano również model oparty o aproksymację dyfuzyjną [12], który w porównaniu do metody aproksymacji Fluid-Flow dostarcza bardziej szczegółowe informacje. Aproksymacja dyfuzyjna pozwala na uzyskanie informacji o rozkładzie kolejki oraz umożliwia rozpatrywanie dwóch parametrów ruchu sieciowego: natężenia i wariancji. Powoła to na lepsze dopasowanie modeli do rzeczywistego ruchu sieciowego. Obie metody pozwalają na analizę zachowania nie tylko w pojedynczych węzłach, natomiast aproksymacja dyfuzyjna pozwala na rozpatrywanie bardziej złożonych konfiguracji sieci. Wyniki przedstawione w tej pracy pokazują zgodność rezultatów uzyskanych metodami analitycznymi i symulacyjnymi. Dodatkowo modele symulacyjne pozwoliły na uzyskanie pewnych rezultatów czasowych, w tym średniego czasu pobytu pakietu w kolejce. Metody analityczne pozwoliły natomiast otrzymać rezultaty, które byłyby trudno osiągalne w przypadku stosowania jedynie metod symulacyjnych [13].

Obydwa modele pozwalają na weryfikację średniej długości kolejki oraz liczby odrzuconych pakietów. Uzyskanie odpowiedzi od kontrolera  $PI^\alpha D^\beta$  niecałkowitego rzędu wiąże się z obliczeniami o dużej złożoności, tj. wykonaniem operacji całkowania i różniczkowania niecałkowitych rzędów. W związku z tym, w celu przeprowadzenia badań mających umożliwić przyspieszenie czasu obliczeń odpowiedzi kontrolera  $PI^\alpha D^\beta$ , w ramach pracy zaproponowano implementację tego rozwiązania w strukturach GPU.

Głównym celem pracy jest ocena zastosowania koncepcji mechanizmu aktywnego zarządzania pakietami AQM, w oparciu o kontroler  $PI^\alpha D^\beta$  niecałkowitych rzędów do zarządzania ruchem pakietów w węzle sieciowym. W związku z tym, tezę niniejszej rozprawy zdefiniowano w sposób następujący:

**Teza 1.** *Oparcie mechanizmu aktywnego zarządzania kolejką (AQM) o kontroler  $PI^\alpha D^\beta$  niecałkowitych rzędów może pozwolić na zmniejszenie średniej zajętości kolejki, a przez to na zwiększenie przepustowości transmisji danych w sieci Internet.*

## Rozdział 2

# Rachunek różniczkowy i różnicowy ułamkowego rzędu

Niniejszy rozdział przedstawia cechy i własności układów ciągłych oraz dyskretnych, które przytoczono w celu umożliwienia właściwej interpretacji pojęć koniecznych do zrozumienia całości pracy. W podrozdziałach zawarto listingi zawierające implementację poszczególnych definicji w języku Python, z wykorzystaniem bibliotek NumPy oraz SciPy, które stosowane są do pracy z tablicami, a także do przeprowadzania obliczeń naukowych oraz technicznych. Wykresy i wizualizacje stworzono natomiast z wykorzystaniem biblioteki Matplotlib.

W wyniku rozwoju rachunku różniczko-całki jaki nastąpił w ostatnich latach, opracowanych zostało wiele metod, które pozwalają na aproksymowanie operacji różniczkowania oraz całkowania niecałkowitego rzędu. Umożliwiło to wykorzystanie ułamkowego rachunku różniczkowego w różnych obszarach nauki, w tym w teorii sterowania, w stosowanych regulatorach, a także przy modelowaniu obiektów z wykorzystaniem równań niecałkowitych rzędów. Spowodowało to również znaczący wzrost zainteresowania regulatorami PID niecałkowitych rzędów.

Dotychczas przeprowadzone badania teoretyczne dowiodły, że w wielu przypadkach algorytmy wykorzystujące rachunek różniczkowy niecałkowitych rzędów działają znacznie lepiej od ich odpowiedników opartych o klasyczny rachunek całkowy rzędów [14, 15]. Znanym problemem związanym z algorytmami wykorzystującymi różniczkę i całki ułamkowych rzędów jest większa złożoność obliczeniowa, pamięciowa i czasowa takich rozwiązań. Z tego też powodu rachunek różniczko-całki niecałkowitych rzędów do tej pory był stosunkowo rzadko wykorzystywany w rzeczywistych aplikacjach.

W pracy tej wykorzystano model dyskretny, wraz z odpowiedzią kontrolerów  $PI^\alpha$  oraz  $PI^\alpha D^\beta$  niecałkowitych rzędów, która stosowana jest jako funkcja prawdopodobieństwa odrzucenia pakietów. W rozdziale tym przedstawiono najpopularniejszą definicję różnicy niecałkowitego rzędu, która jest uogólnieniem podstawowej definicji różnicy niecałkowitego rzędu Grünwalda-Letnikowa. W celu lepszego wyjaśnienia podstaw teoretycznych pojęć wykorzystanych w ramach niniejszej rozprawy, omówiono najważniejsze definicje rachunku różniczkowego oraz różnicowego ułamkowego rzędu. W związku z tym w podrozdziale 2.1 przedstawiono podstawy teoretyczne funkcji  $\Gamma(x)$ , natomiast w podrozdziale 2.2 zaprezentowano własności funkcji Mittag-Lefflera. Podrozdział 2.3 zawiera najważniejsze cechy dynamicznych układów ciągłych, w tym definicję różniczko-całki niecałkowitego rzędu. Natomiast w podrozdziale 2.4 opisano własności układów dyskretnych oraz definicję różnicy niecałkowitego rzędu.

## 2.1 Funkcja $\Gamma(x)$

Własności funkcji  $\Gamma(x)$ , która jest uogólnieniem silni na argumenty rzeczywiste, będą stosowane w kolejnych rozdziałach. Posiada ona dwie definicje - całkową oraz w postaci granicy.

### 1. Definicja całkowa (Eulera):

$$\Gamma(x) = \int_0^{\infty} e^{-t} t^{x-1} dt \quad (2.1)$$

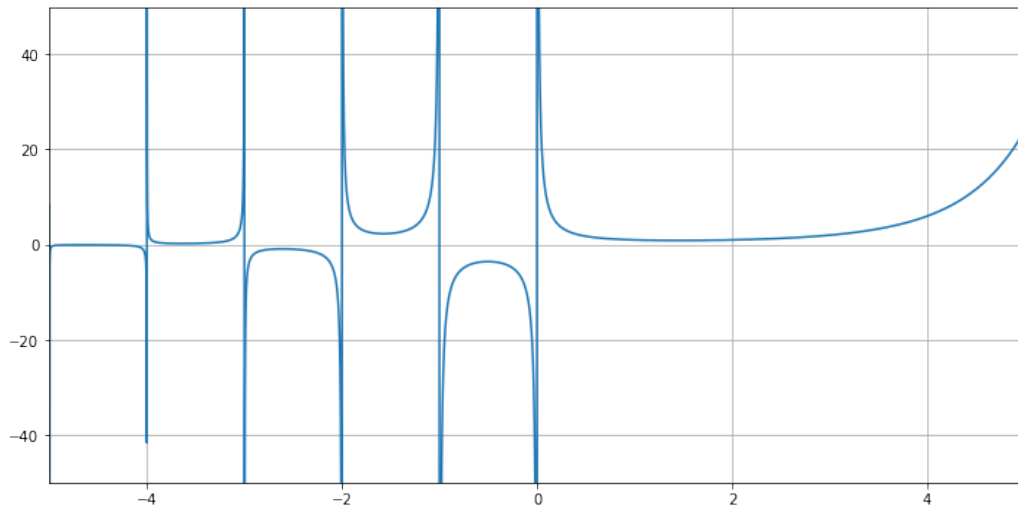
gdzie  $Re(x) > 0$ .

### 2. Definicja w postaci granicy:

$$\Gamma(x) = \lim_{n \rightarrow \infty} \frac{n! n^x}{x(x+1)\dots(x+n)} \quad (2.2)$$

gdzie  $x \in C$ , a podstawowa własność funkcji  $\Gamma(x)$  to:

$$\Gamma(x+1) = x\Gamma(x) \quad (2.3)$$



Rysunek 2.1: Przebieg funkcji  $\Gamma(x)$ .

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.special import gamma

x = np.arange(-20, 100, 0.001)
plt.figure(figsize=(12,6))
plt.plot(x, gamma(x))
plt.ylim(-50, 50); plt.xlim(-5, 5); plt.grid()
```

Listing 1. Kod w języku Python generujący wykres przebiegu funkcji  $\Gamma(x)$ .

## 2.2 Funkcja Mittag-Lefflera

Funkcja Mittag-Lefflera (funkcja zmiennej zespolonej) wykorzystywana jest do rozwiązywania równań różniczkowych, jako uogólnienie funkcji  $e^{s_i t}$  dla równań niecałkowitego rzędu. Funkcja ta istnieje w postaci jednoparametrowej oraz dwuparametrowej, jako rozszerzenie definicji jednoparametrowej [16].

### 1. Definicja jednoparametrowej funkcji Mittag-Lefflera:

$$E_{\alpha}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + 1)} \quad (2.4)$$

gdzie dla  $\alpha = 1$  wynikiem jest funkcja eksponencjalna:

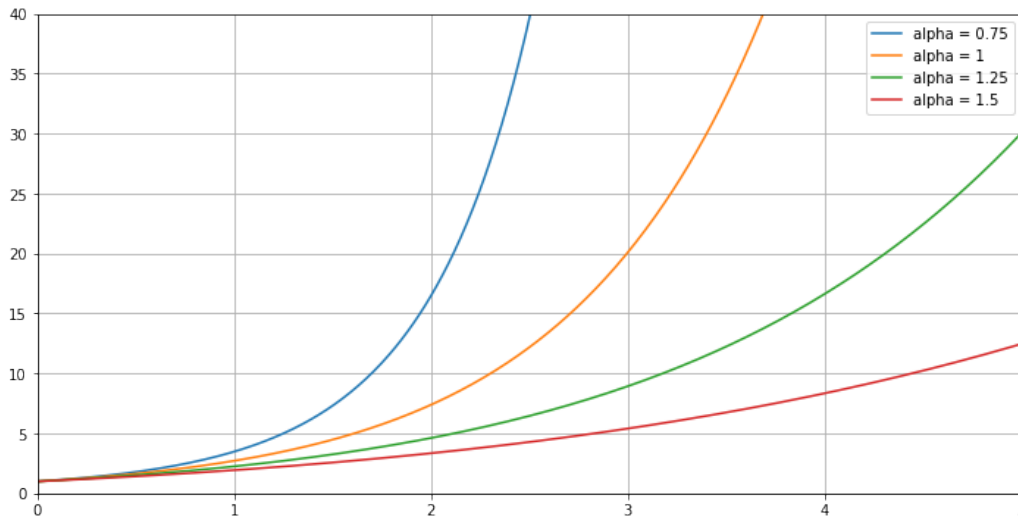
$$E_1(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(k+1)} = \sum_{k=0}^{\infty} \frac{z^k}{k!} = e^z \quad (2.5)$$

Pełny opis i własności jednoparametrowej funkcji Mittaga-Lefflera przedstawiono w pracach [17, 18].

## 2. Definicja dwuparametrowej funkcji Mittaga-Lefflera:

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)} \quad (2.6)$$

gdzie dla  $\beta = 1$  rezultatem jest jednoparametrowa funkcja  $E_{\alpha}(x)$ .



Rysunek 2.2: Przebieg funkcji Mittaga-Lefflera dla wybranych wartości parametru  $\alpha$ .

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.special import gamma

def MittagLeffler(z, alpha, beta):
    k = np.arange(50).reshape(50, 1)
```

```

E = z**k / gamma(alpha*k + beta)
return np.sum(E, axis=0)

x = np.arange(-30, 30, 0.01)
plt.figure(figsize=(12,6))
plt.plot(x, MittagLeffler(x, 0.75, 1),
         label="alpha = 0.75")
plt.plot(x, MittagLeffler(x, 1, 1),
         label="alpha = 1")
plt.plot(x, MittagLeffler(x, 1.25, 1),
         label="alpha = 1.25")
plt.plot(x, MittagLeffler(x, 1.5, 1),
         label="alpha = 1.5")
plt.legend()
plt.ylim(0, 40); plt.xlim(0, 5); plt.grid()

```

Listing 2. Kod w języku Python obliczający i generujący wykres przebiegu funkcji Mittag-Lefflera.

## 2.3 Układy ciągłe

W poniższym podrozdziale ujednoczona zostanie terminologia różniczki i całki, w celu uzyskania definicji różniczko-całki. W przypadku, gdy ( $\alpha > 0$ ) otrzymana zostanie pochodna, dla ujemnego rzędu ( $\alpha < 0$ ) uzyskana zostanie całka, natomiast gdy dla ( $\alpha = 0$ ) zwrócona będzie ta sama funkcja [16]. W pierwszej kolejności przeanalizowane zostaną własności Riemanna-Liouville'a (R-L), które wynikają z rozszerzenia reguły Cauchy'ego dla wielokrotnego całkowania. W drugim przypadku rozważana będzie definicja Grunwalda-Letnikowa (G-L), wynikająca z pochodnej wielokrotnej całkowego rzędu [16]. Uwzględnić należy fakt jednoznaczności definicji Riemanna-Liouville'a i Grunwalda-Letnikowa przy odpowiedniej klasie różniczkowalności funkcji [19]. Równoznaczne definicje różniczko-całki ułamkowego rzędu otrzymać można zarówno poprzez uogólnienie definicji całki, jak też pochodnej.

### 2.3.1 Definicja Riemanna-Liouville'a

Unifikując całkę ułamkowego rzędu zgodnie z założeniami [20], jej definicję należy rozpocząć od wzoru na całkowanie wielokrotne:

$${}_a I_x^n f(x) = \int_a^x du_1 \int_a^{u_1} du_2 \dots \int_a^{u_{n-1}} f(u_n) du_n = \frac{1}{(n-1)!} \int_a^x (x-u)^{n-1} f(u) du \quad (2.7)$$

gdzie  $n \in N$  to krotność całkowania,  $(a, x)$  to przedział całkowania funkcji  $f(u)$ . Bazując na zależności  $(n-1)! = \Gamma(n)$  możliwe jest uogólnienie wzoru 2.7 do  $n \in R$ . Wyznaczamy w ten sposób całkę ułamkowego rzędu zgodnie z regułą Riemanna-Liouville'a.

$${}_a I_t^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t-\tau)^{\alpha-1} f(\tau) d\tau \quad (2.8)$$

gdzie  $n \in R^+$  to rząd całkowania w granicach  $(a, x)$  funkcji  $f(x)$ . Rozszerzając powyższe na  $\alpha < 0$  uzyskać można definicję różniczko-całki ułamkowego rzędu:

$${}_a D_t^\alpha = \frac{d^k}{dt_a} I_t^{(\alpha-k)} f(t) = \frac{1}{\Gamma(k-\alpha)} \frac{d^k}{dt} \int_a^t (t-\tau)^{k-\alpha-1} f(\tau) d\tau \quad (2.9)$$

gdzie  $k-1 \leq \alpha \leq k$ .

Różniczko-całka niecałkowitego rzędu po przekształceniu Laplace'a ma postać:

$$L[{}_0 D_t^\alpha f(t)] = \begin{cases} s^\alpha F(s) & : n < 0 \\ s^\alpha F(s) - \sum_{k=0}^{j-1} s^k {}_0 D_x^{\alpha-k-1} f(0) & : n > 0, \quad j-1 < \alpha \leq j \in N \end{cases} \quad (2.10)$$

gdzie warunki początkowe są pochodnymi ułamkowego rzędu w punkcie  $t = 0$ .

**Twierdzenie:**  ${}_a D_t^\alpha$  to operator liniowy, zachodzi więc zależność:

$${}_a D_t^\alpha (\lambda f(t) + \mu g(t)) = \lambda {}_a D_t^\alpha f(t) + \mu {}_a D_t^\alpha g(t) \quad (2.11)$$

**Dowód:**

$$\begin{aligned} {}_a D_t^\alpha (\lambda f(t) + \mu g(t)) &= \frac{1}{\Gamma(k-\alpha)} \frac{d^k}{dt} \int_a^t (t-\tau)^{k-\alpha-1} (\lambda f(\tau) + \mu g(\tau)) d\tau = \\ &= \frac{1}{\Gamma(k-\alpha)} \frac{d^k}{dt} \int_a^t (t-\tau)^{k-\alpha-1} \lambda f(\tau) d\tau + \frac{1}{\Gamma(k-\alpha)} \frac{d^k}{dt} \int_a^t (t-\tau)^{k-\alpha-1} \mu g(\tau) d\tau = \\ &= \lambda {}_a D_t^\alpha f(t) + \mu {}_a D_t^\alpha g(t) \end{aligned} \quad (2.12)$$

gdzie  $\alpha$  to rząd całkowania, natomiast  $(x_0, x)$  to granica całkowania. Wszystkie własności definicji Riemanna-Liouville'a opisane zostały szczegółowo w [19, 21, 22].



### 2.3.2 Definicja Caputo

Wykorzystanie definicji Riemanna-Liouville'a skutkuje dylematem dotyczącym określenia warunków początkowych, zdefiniowanych za pomocą pochodnych niecałkowitego rzędu [19]. Definicja Caputo znajduje w tym miejscu swoje zastosowanie w celu rozwiązania wyżej przytoczonego problemu. Jej pełną treść przedstawiono w pracach [23, 24].

Różniczko-całka ułamkowego rzędu na podstawie definicji Caputo [19]:

$${}^C D_t^\alpha f(t) = \frac{1}{\Gamma(\alpha - n)} \int_a^t \frac{f^{(n)}(\tau)}{(t - \tau)^{\alpha+1-n}} d\tau \quad (2.13)$$

gdzie  $(n - 1 < \alpha < n)$ .

Na podstawie powyższej definicji przekształcenie Laplace'a ma postać:

$$L[{}_0 D_x^\alpha f(x)] = s^\alpha F(s) - \sum_{k=0}^{n-1} s^{\alpha-k-1} f^{(k)}(0) \quad (2.14)$$

gdzie  $(n - 1 < \alpha \leq n)$ .

Otrzymane warunki początkowe dla przekształcenia Laplace'a zgodnie z definicją Caputo są pochodnymi całkowitego rzędu. Definicja ta posiada więc większe znaczenie praktyczne. Kolejną różnicą twierdzenia Caputo względem definicji Riemanna-Liouville'a jest wynik różniczkowania funkcji stałej dla  $\alpha > 0$  rezultat to 0, gdy dla definicji  $R-L$ , wynik ten występuje jedynie dla całkowitych rzędów. Dla ułamkowych rzędów i skończonej dolnej granicy  $\alpha$  pochodna ma postać:

$${}_0 D_t^\alpha C = \frac{C t^\alpha}{\Gamma(1 - \alpha)} \quad (2.15)$$

Definicje Caputo i R-L są równoważne dla dolnej granicy  $\alpha = -\infty$  [19].

**Twierdzenie:**  ${}^C D_t^\alpha$  to operator liniowy, zachodzi więc zależność:

$${}^C D_t^\alpha (\lambda f(t) + \mu g(t)) = \lambda {}^C D_t^\alpha f(t) + \mu {}^C D_t^\alpha g(t) \quad (2.16)$$

**Dowód:**

$$\begin{aligned} {}^C D_t^\alpha (\lambda f(t) + \mu g(t)) &= \frac{1}{\Gamma(\alpha - n)} \int_a^t \frac{\frac{d^n}{dt^n} (\lambda f(\tau) + \mu g(\tau))}{(t - \tau)^{\alpha+1-n}} d\tau = \\ &= \frac{1}{\Gamma(\alpha - n)} \int_a^t \frac{\frac{d^n}{dt^n} (\lambda f(\tau))}{(t - \tau)^{\alpha+1-n}} d\tau + \frac{1}{\Gamma(\alpha - n)} \int_a^t \frac{\frac{d^n}{dt^n} (\mu g(\tau))}{(t - \tau)^{\alpha+1-n}} d\tau = \\ &= \lambda {}^C D_t^\alpha f(t) + \mu {}^C D_t^\alpha g(t) \end{aligned} \quad (2.17)$$

Wszystkie własności opisane zostały w [19, 21, 22].

### 2.3.3 Definicja Grünwalda-Letnikowa

Różniczko-całkę niecałkowitego rzędu opisać można również za pomocą definicji Grünwalda-Letnikowa, gdzie różniczki:

- pierwszego rzędu definiuje się następująco:

$$\frac{df}{dt} = \lim_{h \rightarrow 0} \frac{f(t) - f(t-h)}{h} \quad (2.18)$$

- drugiego rzędu występują jako:

$$\frac{d^2 f}{dt^2} = \lim_{h \rightarrow 0} \frac{\frac{df}{dt} f(t) - \frac{df}{dt} f(t-h)}{h} = \lim_{h \rightarrow 0} \frac{f(t) - 2f(t-h) + f(t-2h)}{h^2} \quad (2.19)$$

- różniczkę dowolnego całkowitego rzędu przedstawia się za pomocą wzoru:

$$\frac{d^n f}{dt^n} = \lim_{h \rightarrow 0} \frac{1}{h^n} \sum_{r=0}^n (-1)^r \binom{n}{r} f(t-rh) \quad (2.20)$$

W przypadku, gdy  $\binom{n}{r} = 0$ , dla  $r > n$ , za pomocą zależności uogólniającej symbol Newtona na liczby rzeczywiste, wyrażenie to można przekształcić w celu przedstawienia różniczko-całki ułamkowego rzędu:

$$\frac{d^\alpha f}{dt^\alpha} = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{r=0}^n (-1)^r \binom{\alpha}{r} f(t-rh) \quad (2.21)$$

**Twierdzenie:**  ${}_a D_t^\alpha$  to operator liniowy, zachodzi więc zależność:

$${}_a D_t^\alpha (\lambda f(t) + \mu g(t)) = \lambda {}_a D_t^\alpha f(t) + \mu {}_a D_t^\alpha g(t) \quad (2.22)$$

**Dowód:**

$$\begin{aligned} {}_a D_t^\alpha (\lambda f(t) + \mu g(t)) &= \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{r=0}^n (-1)^r \binom{\alpha}{r} (\lambda f(t-rh) + \mu g(t-rh)) = \\ &= \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{r=0}^n (-1)^r \binom{\alpha}{r} (\lambda f(t-rh)) + \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{r=0}^n (-1)^r \binom{\alpha}{r} (\mu g(t-rh)) = \\ &= \lambda {}_a D_t^\alpha f(t) + \mu {}_a D_t^\alpha g(t) \end{aligned} \quad (2.23)$$

Wszystkie własności opisane zostały w [19, 21, 22].

### 2.3.4 Operatory różniczkowania ułamkowego rzędu

Gdy warunki początkowe dla funkcji i jej pochodnych są zerowe, niezależnie od zastosowanej definicji różniczko-całki niecałkowitego rzędu, zachodzi poniższe twierdzenie składania pochodnych:

$${}_0D_t^\alpha({}_0D_t^\beta f(t)) = {}_0D_t^\beta({}_0D_t^\alpha f(t)) = {}_0D_t^{\alpha+\beta} f(t) \quad (2.24)$$

Gdy warunki początkowe nie są zerowe, twierdzenie to jest bardziej złożone (na przykładzie definicji R-L):

$${}_aD_t^\alpha({}_aD_t^\beta f(t)) = {}_aD_t^\beta({}_aD_t^\alpha f(t)) = {}_aD_t^{\alpha+\beta} f(t) - \sum_{j=1}^m [{}_aD_t^{\beta-j} f(t)]_{t=a} \frac{(t-a)^{\alpha-j}}{\Gamma(\alpha-1-j)} \quad (2.25)$$

Wszystkie własności opisane zostały w [19, 21, 22, 25].

### 2.3.5 Różniczkowanie niecałkowitego rzędu

W poniższej części rozprawy opisane zostaną analityczne metody obliczania różniczki niecałkowitego rzędu. Pochodna rzędu  $n = 0.5$  funkcji  $1(t)$  z wykorzystaniem definicji Riemanna-Liouville'a:

$$\begin{aligned} d^{0.5}1(t) &= \frac{d}{dt} \left[ \frac{1}{\Gamma(0.5)} \int_0^t (t-u)^{-0.5} du \right] = \\ &= \frac{1}{\Gamma(0.5)} \frac{d}{dt} [2(t-u)^{0.5}]_0^t = \frac{1}{\Gamma(0.5)} \frac{d}{dt} 2(t)^{0.5} = \frac{1}{\Gamma(0.5)} t^{-0.5} \end{aligned} \quad (2.26)$$

Pochodna rzędu  $\alpha$  funkcji stałej z wykorzystaniem definicji Riemanna-Liouville'a:

$$\begin{aligned} \frac{d^\alpha}{dt^\alpha} 1(t) &= \frac{1}{\Gamma(k-\alpha)} \frac{d^k}{dt^k} \left[ \int_0^t (t-u)^{k-\alpha-1} du \right] = \\ &= \frac{1}{\Gamma(k-\alpha)} \frac{d^k}{dt^k} \left[ \frac{1}{k-\alpha} (t-u)^{k-\alpha} \right]_0^t = \\ &= \frac{1}{\Gamma(k-\alpha)} \frac{d^k}{dt^k} \frac{1}{k-\alpha} t^{k-\alpha} = \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{\Gamma(k-\alpha)} \frac{1}{\alpha-k} (k-\alpha)(k-\alpha-1)\dots(1-\alpha)t^{-\alpha} = \\
&= \frac{(1-\alpha)\dots(k-\alpha)}{\Gamma(1-\alpha)(1-\alpha)\dots(k-\alpha)} t^{-\alpha} = \\
&= \frac{t^{-\alpha}}{\Gamma(1-\alpha)} \tag{2.27}
\end{aligned}$$

## 2.4 Układy dyskretne

W niniejszym podrozdziale omówiono tematykę rachunku różnicowego niecałkowitego rzędu dla układów dyskretnych oraz opisano najpopularniejszą definicję różnicy niecałkowitego rzędu, która jest uogólnieniem definicji Grünwalda-Letnikowa dla ustalonego kroku  $h$ . Definicja ta stanowi zarazem podstawę teoretyczną metody zaprezentowanej w niniejszej rozprawie. Wynika to z faktu, że obliczanie wartości funkcji prawdopodobieństwa odrzucenia pakietu z kolejki jest zdarzeniem dyskretnym, ponieważ odbywa się w momencie nadejścia pakietu do kolejki. Definicje różnic określonych rzędów przedstawia się następująco:

- pierwszego rzędu występuje jako:

$$\Delta x_k = x_k - x_{k-1} \tag{2.28}$$

- drugiego rzędu przedstawia się za pomocą wzoru:

$$\Delta^2 x_k = x_k - 2x_{k-1} + x_{k-2} \tag{2.29}$$

- dowolnego rzędu określona jest jako:

$$\Delta^n x_k = \sum_{r=0}^n (-1)^r \binom{n}{r} x_{k-r} \tag{2.30}$$

Biorąc pod uwagę, że  $\binom{n}{r}$  jest równy 0, dla  $r > n$  można zapisać definicję różnicy niecałkowitego rzędu w następujący sposób:

$$\Delta^n x_k = \sum_{j=0}^n (-1)^j \binom{n}{j} x_{k-j} \tag{2.31}$$

gdzie  $n \in R$  to niecałkowity rząd,  $k \in N$  to numer próbki obliczanej różnicy, a  $x_k$  to różnicowana funkcja dyskretna.

Natomiast współczynnik  $\binom{n}{r}$  obliczany jest za pomocą wzoru:

$$\binom{n}{j} = \begin{cases} 1 & : j = 0 \\ \frac{n(n-1)\dots(n-j+1)}{j!} & : j > 0 \end{cases} \quad (2.32)$$

**Przykłady obliczania różnicy niecałkowitego rzędu:**

Dla różnicy pierwszego rzędu, gdy  $n = 1$  jedynie dwa pierwsze współczynniki nie są zerowe:

$$\Delta^1 x_k = 1x_k - 1x_{k-1} + 0x_{k-2} + 0x_{k-3} \dots \quad (2.33)$$

Dla  $n = -1$  obliczamy sumę wszystkich próbek, a więc dyskretny odpowiednik całki pojedynczej:

$$\Delta^{-1} x_k = 1x_k + 1x_{k-1} + 1x_{k-2} + 1x_{k-3} \dots \quad (2.34)$$

Dla  $n = 0.5$  wyznaczamy różnicę rzędu 0.5, czyli sumę ważoną wszystkich próbek:

$$\Delta^{0.5} x_k = 1x_k - 0.5x_{k-1} - 0.125x_{k-2} - 0.0625x_{k-3} \dots \quad (2.35)$$

Dla  $n = -0.5$  wyznaczamy:

$$\Delta^{-0.5} x_k = 1x_k + 0.5x_{k-1} + 0.375x_{k-2} + 0.3125x_{k-3} \dots \quad (2.36)$$

Dla  $n = -1.2$  wyznaczamy:

$$\Delta^{-1.2} x_k = 1x_k + 1.2x_{k-1} + 1.32x_{k-2} + 1.408x_{k-3} \dots \quad (2.37)$$

Dla  $n = -0.8$  wyznaczamy:

$$\Delta^{-0.8} x_k = 1x_k + 0.8x_{k-1} + 0.72x_{k-2} + 0.672x_{k-3} \dots \quad (2.38)$$

Dla  $n = -0.4$  wyznaczamy:

$$\Delta^{-0.4} x_k = 1x_k + 0.4x_{k-1} + 0.28x_{k-2} + 0.224x_{k-3} \dots \quad (2.39)$$

### 2.4.1 Różnica niecałkowitego rzędu zapisana macierzowo

W literaturze [26] dostępna jest różnica niecałkowitego rzędu zapisana jako macierz współczynników.

Przyjmując, że:

$$b_i^n = (-1)^i \binom{n}{i} \quad (2.40)$$

definicja różnicy niecałkowitego rzędu określona jest wtedy jako:

$$\Delta^n x_k = \sum_{j=0}^k b_j^{(n)} x_{k-j} \quad (2.41)$$

Możliwe jest wtedy oznaczenie:

$$\begin{bmatrix} \Delta^n f_k \\ \Delta^n f_{k-1} \\ \Delta^n f_{k-2} \\ \Delta^n f_{k-3} \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & b_1^{(n)} & b_2^{(n)} & b_3^{(n)} & \dots \\ 0 & 1 & b_1^{(n)} & b_2^{(n)} & \dots \\ 0 & 0 & 1 & b_1^{(n)} & \dots \\ 0 & 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} f_k \\ f_{k-1} \\ f_{k-2} \\ f_{k-3} \\ \vdots \end{bmatrix}$$

Macierz kwadratowa opisana jest jako:

$$B^{(n)} = \begin{bmatrix} 1 & b_1^{(n)} & b_2^{(n)} & b_3^{(n)} & \dots \\ 0 & 1 & b_1^{(n)} & b_2^{(n)} & \dots \\ 0 & 0 & 1 & b_1^{(n)} & \dots \\ 0 & 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

dla której wyznacznik jest różny od 0 oraz zawsze posiada odwrotność, przedstawioną jako:

$$A^{(n)} = [B^{(n)}]^{-1} = \begin{bmatrix} 1 & a_1^{(n)} & a_2^{(n)} & a_3^{(n)} & \dots \\ 0 & 1 & a_1^{(n)} & a_2^{(n)} & \dots \\ 0 & 0 & 1 & a_1^{(n)} & \dots \\ 0 & 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

gdzie:

$$a_i^{(n)} = \frac{n(n-1)\dots(n-i+1)}{i!} \quad (2.42)$$

Wynikiem czego jest:

$$\begin{bmatrix} 1 & a_1^{(n)} & a_2^{(n)} & a_3^{(n)} & \dots \\ 0 & 1 & a_1^{(n)} & a_2^{(n)} & \dots \\ 0 & 0 & 1 & a_1^{(n)} & \dots \\ 0 & 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \Delta^n f_k \\ \Delta^n f_{k-1} \\ \Delta^n f_{k-2} \\ \Delta^n f_{k-3} \\ \vdots \end{bmatrix} = \begin{bmatrix} f_k \\ f_{k-1} \\ f_{k-2} \\ f_{k-3} \\ \vdots \end{bmatrix}$$

gdzie między macierzami  $A^n$  i  $B^n$  spełnione są zależności:

$$A^{(n_1+n_2)} = A^{(n_1)} A^{(n_2)}$$

$$B^{(n_1+n_2)} = B^{(n_1)} B^{(n_2)}$$

## 2.4.2 Różnica niecałkowitego rzędu - implementacja

Różnica niecałkowitego rzędu zawiera sumowanie wszystkich kolejnych próbek aż do aktualnej, wraz z ich współczynnikami. Gdy czas pracy danego układu dąży do nieskończoności, wtedy analogicznie rośnie liczba branych pod uwagę próbek. Prowadzi to do problemów obliczeniowych, wynikających między innymi z zajętości pamięci.

Przystępując do praktycznej implementacji różnicy niecałkowitego rzędu należy wziąć pod uwagę fakt, że współczynnik  $\binom{n}{j}$  maleje dla odległych próbek. Skutkuje to ich niewielkim wpływem na końcowy rezultat. Można więc ograniczyć ilość sumowanych próbek do pewnej zadanej maksymalnej wartości  $M$ , będącej zarazem długością implementacji.

Dla tak zdefiniowanych założeń, definicja różnicy niecałkowitego wyrażona jest następująco:

$$\Delta^n x_k = \begin{cases} \sum_{j=0}^k (-1)^j \binom{n}{j} x_{k-j} & : k < L \\ \sum_{j=0}^M (-1)^j \binom{n}{j} x_{k-j} & : k \geq M \end{cases} \quad (2.43)$$

Praktyka pokazuje, że optymalne rezultaty osiągnąć są zazwyczaj dla  $M$  rzędu kilkuset.

## Rozdział 3

# Kontroler ułamkowego rzędu $PI^\alpha D^\beta$ jako mechanizm aktywnego zarządzania kolejką

Pochodne i całki ułamkowego rzędu (ang. Fractional Order Derivatives and Integrals) są naturalnym rozszerzeniem całek i pochodnych. Różniczki niecałkowitego rzędu umożliwiają lepszą i bardziej precyzyjną kontrolę procesów dynamicznych, pozwalając na uwzględnienie pamięci danego procesu. Istnieje kilka definicji związanych z operatorami niecałkowitego rzędu. Najpopularniejsze definicje procesów ciągłych zostały opisane w rozdziale 2.

Pierwsze propozycje wykorzystania kontrolera PI jako mechanizmu aktywnego zarządzania kolejką przedstawili autorzy z Uniwersytetu Massachusetts w 2001 oraz 2002 roku [27, 28]. Prace te doczekały się licznych kontynuacji w kolejnych latach, kiedy to autorzy z całego świata tworzyli własne modele symulacyjne, a następnie implementacje zaproponowanego rozwiązania w rzeczywistych sieciach komputerowych. Badania te miały na celu ocenę mechanizmu PI oraz porównania jego wydajności względem klasycznych mechanizmów aktywnego zarządzania kolejką [29, 30, 31, 32, 33].

W wyniku zainteresowania rachunkiem różniczko-całki i jego stosowaniem zarówno w ciągłych, jak i dyskretnych układach dynamicznych, w ciągu ostatnich kilku lat w literaturze pojawiło się wiele badań dowodzących zalet stosowania mechanizmów ułamkowego rzędu. Autorzy prac [34, 35, 36] w swoich badaniach wykazali, że regulatory niecałkowitego rzędu osiągają lepsze wyniki niż najlepsze klasyczne mechanizmy o rzędach rzeczywistych. W artykule [37] opisano zastosowanie regulatora PID o ułamkowym rzędzie całkowania do stero-



wania temperaturą w pomieszczeniu izotermicznym zakładu farmaceutycznego. Rezultaty przeprowadzonych prac symulacyjnych jednoznacznie wskazują, że zaproponowany regulator zapewnia większą dokładność regulacji od klasycznego kontrolera PID. Autorzy artykułu [38] dowiedli również, że niecałkowity mechanizm PID pozwala uzyskać zwiększenie stabilności pracy systemu.

Pierwszą implementację kontrolera niecałkowitego rzędu PI, którą zastosowano jako mechanizm aktywnego zarządzania kolejką w modelu aproksymacji Fluid-Flow, zaproponowali w 2014 roku w swojej pracy autorzy Krajewski i Viaro [39].

W pracach [1, 40, 41, 42, 43] przedstawiono dalszy rozwój koncepcji związanej z wykorzystaniem odpowiedzi regulatora niecałkowitego rzędu  $PI^\alpha D^\beta$ , jako mechanizmu aktywnego zarządzania kolejką.

Obliczanie wartości funkcji prawdopodobieństwa odrzucenia pakietu z kolejki w oparciu o model kontrolera  $PI^\alpha D^\beta$  jest zdarzeniem dyskretnym, ponieważ odbywa się w momencie nadejścia pakietu do kolejki. Z tej też przyczyny rozważania zaprezentowane w ramach niniejszej pracy oparte są o wykorzystanie definicji dyskretnego operatora ułamkowego Grünwalda-Letnikova [21, 44]. Stanowi ona zarazem uogólnienie tradycyjnej definicji różnicy całkowitego rzędu na rząd niecałkowity, której pełne wyprowadzenie zaprezentowano w podrozdziale 2.4 niniejszej pracy.

Dla danej sekwencji  $f_0, f_1, \dots, f_j, \dots, f_k$  [45, 19]:

$$\Delta^\alpha f_k = \sum_{j=0}^k (-1)^j \binom{\alpha}{j} f_{k-j} \quad (3.1)$$

gdzie  $\alpha \in R$  to niecałkowity rząd całkowania,  $f_k$  to zróżnicowana funkcja dyskretna, a  $\binom{\alpha}{j}$  jest uogólnionym symbolem Newtona, który zdefiniowano w sposób następujący:

$$\binom{\alpha}{j} = \begin{cases} 1 & \text{for } j = 0 \\ \frac{\alpha(\alpha-1)(\alpha-2)\dots(\alpha-j+1)}{j!} & \text{for } j = 1, 2, \dots \end{cases} \quad (3.2)$$

Dla  $\alpha = 1$  otrzymano wzór na różnicę pierwszego rzędu, gdzie jedynie dwa współczynniki mają wartość niezerową.

$$\Delta^1 x_k = 1x_k - 1x_{k-1} + 0x_{k-2} + 0x_{k-3} \dots \quad (3.3)$$

Dla  $\alpha = -1$  otrzymano sumę wszystkich próbek:

$$\Delta^{-1} x_k = 1x_k + 1x_{k-1} + 1x_{k-2} + 1x_{k-3} \dots \quad (3.4)$$

Dla niecałkowitego rzędu całkowania otrzymujemy sumę ważoną wszystkich próbek:

$$\begin{aligned}\Delta^{-1.2}x_k &= 1x_k + 1.2x_{k-1} + 1.32x_{k-2} + 1.408x_{k-3} \dots \\ \Delta^{-0.8}x_k &= 1x_k + 0.8x_{k-1} + 0.72x_{k-2} + 0.672x_{k-3} \dots\end{aligned}\quad (3.5)$$

Prawdopodobieństwo odrzucenia pakietu wyliczane jest za pomocą odpowiedzi kontrolera  $PI^\alpha D^\beta$ :

$$P = \max\{0, -(K_P e_k + K_I \Delta^\alpha e_k + K_D \Delta^\beta e_k)\} \quad (3.6)$$

gdzie  $K_P, K_I, K_D$  są to współczynniki regulatora, odpowiednio proporcjonalny, całkujący i różniczkujący, a  $\alpha, \beta$  są funkcjami rzędu sumowania oraz różnicowania, czyli są parametrami całkowania i różniczkowania niecałkowitego rzędu. Natomiast  $e_k$  jest uchybem regulatora w aktualnym slocie czasowym ( $e_k = Q_k - Q$ ), czyli stanowi różnicę między bieżącym rozmiarem kolejki  $Q_k$ , a jej wartością oczekiwaną  $Q$ .

W poniższym listingu przedstawiono implementację kontrolera  $PI^\alpha D^\beta$  niecałkowitego rzędu w języku Python:

```
class PIDController():

def __init__(self, setPoint, kp, ki, kd, labda,
mi, maxErrors, dCoefficientStartValue = 1,
iCoefficientStartValue = 1):
self.setPoint = setPoint
self.kp = kp
self.kd = kd
self.ki = ki
self.labda = labda
self.mi = mi
self.maxErrors = maxErrors
self.errors = []
self.dCoefficientStartValue = \
dCoefficientStartValue
self.iCoefficientStartValue = \
iCoefficientStartValue
self.createICoefficientsVector()
self.createDCoefficientsVector()
```

```

def addError(self, newError):
    if self.maxErrors > 0 and len(self.errors) \
    >= self.maxErrors - 1:
        del self.errors[-1]
    self.errors.insert(0,newError)

def ni(self, n, i, pom):
    if i == 0:
        return 1.0
    else:
        return pom * (1 - ((1 + n) / i * 1.0))

def createCoefficientsVector(self, n, startValue,
    coefficientsMaxCount):
    coefficients = []
    coefficients.append(self.ni(n, 0, startValue))

    if 0 < coefficientsMaxCount:
        for iter in range(1, coefficientsMaxCount):
            coefficients.append(self.ni(n, iter,
                coefficients[-1]))

    return coefficients

def createICoefficientsVector(self):
    self.iCoefficientsVector =
        self.createCoefficientsVector(self.labda,
            self.iCoefficientStartValue,
            self.maxErrors)

def createDCoefficientsVector(self):
    self.dCoefficientsVector =
        self.createCoefficientsVector(self.mi,
            self.dCoefficientStartValue,
            self.maxErrors)

```

```

def multiplyErrorsWithCoefficientsAndSum(self,
                                         coefficients):
    tmpReturnValue = 0.0

    for i in range(0, min(len(self.errors),
                          len(coefficients))):
        tmpReturnValue \
            += self.errors[i] * coefficients[i]

    return tmpReturnValue

def calculateProportionalTermValue(self):
    tmpReturnValue = 0.0

    if self.kp != 0:
        tmpReturnValue = self.errors[0] * self.kp

    return tmpReturnValue

def calculateDerivativeTermValue(self):
    tmpReturnValue = 0.0

    if self.kd != 0:
        tmpReturnValue
        = self.kd \
            * self.multiplyErrorsWithCoefficientsAndSum
            (self.dCoefficientsVector)

    return tmpReturnValue

def calculateIntegralTermValue(self):
    tmpReturnValue = 0.0

    if self.ki != 0:

```

```

        tmpReturnValue
        = self.ki \
            * self.multiplyErrorsWithCoefficientsAndSum
              (self.iCoefficientsVector)

    return tmpReturnValue

def setControllerCoefficients(self, kp, ki,
                               kd, labda, mi):
    self.kp = kp
    self.kd = kd
    self.ki = ki
    self.labda = labda
    self.mi = mi
    self.createICoefficientsVector()
    self.createDCoefficientsVector()

def update(self, currentValue):
    self.addError(self.setPoint - currentValue)
    tmpPTermValue = \
        self.calculateProportionalTermValue()
    tmpITermValue = \
        self.calculateDerivativeTermValue()
    tmpDTermValue = \
        self.calculateegralTermValue()

    return tmpPTermValue + tmpITermValue \
        + tmpDTermValue

```

Listing 6. Implementacja kontrolera  $PI^\alpha D^\beta$  niecałkowitego rzędu w języku Python.

Efektywność mechanizmu AQM opartego o regulator niecałkowitego rzędu  $PI^\alpha D^\beta$  w dużym stopniu zależy od właściwego doboru parametrów tego kontrolera.

## Rozdział 4

# Analityczne metody ewaluacji mechanizmów aktywnego zarządzania kolejką

W niniejszym rozdziale przedstawione zostały metody, które wykorzystano do oceny zaproponowanego mechanizmu aktywnego zarządzania kolejką, opartego o mechanizm kontrolera niecałkowitego rzędu  $PI^\alpha D^\beta$ . W tym celu, oprócz symulacji zdarzeń dyskretnych zaimplementowanej w środowisku SimPy, zastosowane zostały również rozwiązania oparte o dwie metody analityczne, aproksymację Fluid-Flow oraz aproksymację dyfuzyjną. Aproksymacja Fluid-Flow umożliwia przeprowadzenie oceny współpracy mechanizmów aktywnego zarządzania kolejką z protokołem TCP. Na potrzeby pracy wykorzystano model TCP, który jest pewnym uproszczeniem algorytmu TCP NewReno [46]. Z drugiej strony model ten ignoruje wpływ zależności czasowych oraz pozwala uzyskać jedynie wartości średnie parametrów transmisji. Z tej też przyczyny zdecydowano również o zastosowaniu modelu aproksymacji dyfuzyjnej. Pozwala ona uzyskać bardziej szczegółowe informacje o przebiegu transmisji danych.

### 4.1 Aproksymacja Fluid-Flow

Do modelowania zachowania pakietów w węźle komunikacyjnym zaproponowano metodę aproksymacji Fluid-Flow [11, 47, 48]. Jest ona zwłaszcza odpowiednia do analizy i modelowania ruchu TCP oraz oceny współpracy strumieni TCP z mechanizmami aktywnego zarządzania kolejką. Model ten jest stosunkowo

prosty, ponieważ nie bierze pod uwagę wpływu zależności czasowych (ang. time-out), które są implementowane w protokole TCP. Model mimo swojej prostoty pozwala na uzyskanie kluczowych parametrów sieciowych, takich jak średnia zajętość kolejki czy zmiany okna przeciążenia.

Przedstawione we wzorach 4.1 oraz 4.2 nieliniowe równania różniczkowe definiują rozszerzony model aproksymacji Fluid-Flow dla transmisji z wieloma niezależnymi strumieniami TCP oraz UDP egzystującymi w jednym łączu transmisyjnym [49, 50]. Model prezentowany przez poniższe wzory umożliwia ocenę wydajności realistycznej sieci, obsługującej wiele przepływów heterogenicznych, czyli transmisji z wykorzystaniem różnych protokołów oraz wzorców ruchu [51].

$$\frac{dW_i^{TCP}(t)}{dt} = \frac{1}{R_i(t)} - \frac{W_i^{TCP}(t)}{2} \frac{W_i^{TCP}(t - R_i(t))}{R_i(t - R_i(t))} p(t - R_i(t)) \quad (4.1)$$

$$\frac{dq(t)}{dt} = \sum_{i=1}^N \frac{W_i^{TCP}(t)}{R_i(t)} - C \quad (4.2)$$

gdzie:

$W_i$  - spodziewany rozmiar okna TCP,

$q$  - oczekiwana liczba pakietów w kolejce,

$R_i$  - round-trip time =  $q/C + T_p$ ,

$C$  - przepustowość łącza,

$T_p$  - czas propagacji sygnału,

$N$  - liczba strumieni TCP,

$p$  - prawdopodobieństwo odrzucenia pakietów, obliczane według zaproponowanego mechanizmu aktywnego zarządzania kolejką (AQM).

Model strumieni UDP wyrażony jest za pomocą strumienia o stałej szybkości transmisji (ang. Constant Bitrate, CBR) z określoną liczbą pakietów wysyłanych w jednostce czasu. Szybkość wysyłania  $i$ -tego strumienia UDP jest aproksymowana za pomocą równania:

$$W_i^{UDP}(t) = U \quad (4.3)$$

Maksymalne wartości  $q$  i  $W$  zależą od pojemności bufora oraz maksymalnego rozmiaru okna [1]. Za pomocą tak wyrażonego modelu aproksymacji Fluid-Flow można przedstawić topologię sieci składającej się z wielu stanowisk obsługi, z dowolną liczbą strumieni TCP [50].

## 4.2 Aproksymacja Fluid-Flow dla modelu sieci wielu stanowisk obsługi

Dostosowanie aproksymacji Fluid-Flow do modelu sieci wielu stanowisk obsługi wymaga wprowadzenia dodatkowych oznaczeń do pierwotnie zaproponowanej metody. Za pomocą wektora  $P(x)$  gromadzone są wartości prawdopodobieństwa odrzucenia pakietów przez poszczególne routery znajdujące się w sieci  $V$ . Natomiast w wierszach macierzy  $A$  oznacza się poszczególne przepływy TCP, a w jej kolumnach pojedyncze węzły sieci. W ten sposób po wymnożeniu wektora  $P(x)$  oraz macierzy  $A$  uzyskuje się macierz  $AP$ , za pomocą której oblicza się sumaryczne prawdopodobieństwo utraty pakietów [52].

Definicję aproksymacji Fluid-Flow dla modelu sieci wielu stanowisk obsługi wyraża się więc za pomocą następującego wzoru 4.4:

$$\frac{dW_i(t)}{dt} = \frac{1}{R_i(q(t))} - \frac{W_i(t)}{2} \frac{W_i(t-R)}{R_i(q(t-R))} (1 - \Pi(1 - AP(x)_i)) \quad (4.4)$$

Przekształceniu ulega również sposób obliczania czasu transmisji pakietu w obie strony [52]:

$$R_i(q(t)) = \sum_{v \in V_i} \frac{q_v(t)}{C(v)} + Tp_i \quad (4.5)$$

gdzie przyjmuje się, że  $V_i$  jest zbiorem połączeń pomiędzy węzłami sieci. Pozostałe metody pozwalające wyznaczyć średnią krocącą długość kolejki nie ulegają zmianie.

## 4.3 Aproksymacja dyfuzyjna

Aproksymacja dyfuzyjna od wielu lat jest stosowana w badaniach oceny wydajności systemów i sieci komputerowych. Istnieją prace, które wyjaśniają w jaki



sposób metodę tę można zastosować m.in. do przeprowadzania analiz związanych z zarządzaniem transmisją w sieciach komputerowych [12]. Istnieją również prace, w których model dyfuzyjny wykorzystano do modelowania transmisji z wykorzystaniem protokołu TCP oraz do oceny mechanizmu aktywnego zarządzania kolejką [53], a także przy badaniu wpływu pojemności bufora na jakość transmisji wideo w połączeniach bezprzewodowych [54].

Aproksymacja dyfuzyjna jest metodą analityczną, umożliwiającą przeprowadzenie analizy estymowania rozkładu zajętości kolejki dla stanów ustalonych, podczas transmisji sieciowej. Poprzez rozwiązanie równania aproksymacji dyfuzyjnej [55, 56, 57, 58], z właściwie dobranymi współczynnikami oraz warunkami brzegowymi, uzyskuje się funkcję gęstości procesu dyfuzji. Z kolei funkcja ta pozwala uzyskać rozkład prawdopodobieństwa długości kolejki.

Główna zasada tej metody polega na zastąpieniu procesu dyskretnego  $N(t)$ , który oznacza liczbę klientów w kolejce, za pomocą ciągłego procesu dyfuzji  $X(t)$ . Podobnie jak w przypadku  $N(t)$ , zmiana  $dX(t) = X(t + dt) - X(t)$  ma rozkład normalny o średniej  $\beta dt$  oraz wariancji  $\alpha dt$ , określonej poprzez parametry  $\beta$ ,  $\alpha$  równania dyfuzji [59, 60]:

$$\frac{\partial f(x, t; x_0)}{\partial t} = \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x} \quad (4.6)$$

gdzie  $f(x, t; x_0)$  to funkcja gęstości prawdopodobieństwa procesu dyfuzji [60]:

$$f(x, t; x_0) dx = P[x \leq X(t) < x + dx | X(0) = x_0]. \quad (4.7)$$

Rozwiązanie równania (4.6) umożliwia ocenę rozkładu kolejki w badanym systemie. Metoda ta znajduje zastosowanie, gdy średni czas między zgłoszeniami do systemu oraz średni czas obsługi pojedynczego zgłoszenia zgodny jest z modelami systemów G/G/1 oraz G/G/1/N. W celu uzyskania informacji o zmianie liczby klientów w systemie rozważane są dwa pierwsze momenty rozkładów:

$$E[A] = 1/\lambda, \quad E[B] = 1/\mu, \quad Var[A] = \sigma_A^2, \quad Var[B] = \sigma_B^2. \quad (4.8)$$

Współczynniki kwadratowe wariancji wyrażone są za pomocą:

$$C_A^2 = \sigma_A^2 \lambda^2, \quad C_B^2 = \sigma_B^2 \mu^2, \quad (4.9)$$

a współczynniki równania dyfuzji jako:

$$\beta = \lambda - \mu, \quad \alpha = \sigma_A^2 \lambda^3 + \sigma_B^2 \mu^3 = C_A^2 \lambda + C_B^2 \mu \quad (4.10)$$

### 4.3.1 Model nieskończonej kolejki: G/G/1

Pełne twierdzenie wyprowadzające równanie aproksymacji dyfuzyjnej zawarte jest w twierdzeniach granicznych dla systemu G/G/1 przedstawionych przez Igleharta i Whitte [61, 62, 63, 64]: Jeśli  $\hat{N}_n$  jest serią zmiennych losowych wyprowadzoną z  $N(t)$  [12]:

$$\hat{N}_n = \frac{N(nt) - (\lambda - \mu)nt}{(\sigma_A^2 \lambda^3 + \sigma_B^2 \mu^3) \sqrt{n}} \quad (4.11)$$

wtedy też dany szereg jest słabo zbieżny dla  $\xi$ , gdzie  $\xi(t)$  jest standardowym procesem Wienera, tj. procesem dyfuzji z  $\beta = 0$  i  $\alpha = 1$  pod warunkiem, że  $\rho > 1$ . Oznacza to, że system jest przeciążony i nie znajduje się w stanie równowagi. W przypadku  $\rho = 1$  szereg  $\hat{N}_n$  jest zbieżny do  $\xi_R$ . Proces  $\xi_R(t)$  jest procesem  $\xi(t)$  ograniczonym do  $x > 0$  [64]:

$$\xi_R(t) = \xi(t) - \inf [\xi(u), 0 \leq u \leq t]. \quad (4.12)$$

Proces  $N(t)$  zawsze jest nieujemny, dlatego  $X(t)$  również powinien być ograniczony do wartości  $x \geq 0$ . Odbywa się to poprzez umieszczenie w  $x = 0$  bariery, która uniemożliwia procesowi przejście w ujemną część osi  $x$ .

Jedną z możliwości jest umieszczenie w  $x = 0$  bariery odbijającej [65], która ogranicza proces jedynie do dodatniej osi  $x$  i jest tożsama z:

$$\int_0^\infty f(x, t; x_0) dx = 1$$

oraz:

$$\frac{\partial}{\partial t} \int_0^\infty f(x, t; x_0) dx = \int_0^\infty \frac{\partial f(x, t; x_0)}{\partial t} dx = 0. \quad (4.13)$$

Poprzez zastąpienie funkcji całkującej prawą stroną równania dyfuzji, otrzymujemy warunek brzegowy odpowiadający barierze odbijającej w zerze [12]:

$$\lim_{x \rightarrow 0} \left[ \frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0) \right] = 0. \quad (4.14)$$

Rozwiązanie równania (4.7) z zastosowaniem warunków brzegowych zdefiniowanych w równaniu (4.14) prowadzi do [65]:

$$f(x, t; x_0) = \frac{\partial}{\partial x} \left[ \Phi\left(\frac{x - x_0 - \beta t}{\alpha t}\right) - e^{\frac{2\beta x}{\alpha}} \Phi\left(\frac{x + x_0 + \beta t}{\alpha t}\right) \right], \quad (4.15)$$

gdzie  $\Phi(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$  jest funkcją gęstości prawdopodobieństwa dla rozkładu normalnego.

Dla ( $\beta < 0$ ) system przybliża się do stanu ustalonego:  $\lim_{t \rightarrow \infty} f(x, t; x_0) = f(x)$ , natomiast cząstkowe równanie różniczkowe (4.6) staje się zwykłym równaniem [12]:

$$0 = \frac{\alpha}{2} \frac{d^2 f(x)}{dx^2} - \beta \frac{d f(x)}{dx}$$

z następującym rozwiązaniem:

$$f(x) = -\frac{2\beta}{\alpha} e^{\frac{2\beta x}{\alpha}}. \quad (4.16)$$

Formuła ta pozwala przybliżyć zajętość kolejki w systemie  $G/G/1$  [12]:

$$p(n, t; n_0) \approx f(n, t; n_0), \quad (4.17)$$

gdzie w stanie ustalonym  $p(n) \approx f(n)$  [59]:

$$p(0) \approx \int_0^{0.5} f(x) dx, p(n) \approx \int_{n-0.5}^{n+0.5} f(x) dx, n = 1, 2, \dots, \quad (4.18)$$

Bariera odbijająca w punkcie  $x = 0$  powoduje natychmiastowe odbicie procesu, czyli  $f(0, t; x_0) = 0$ . Niniejsza wersja procesu dyfuzji służy do przybliżenia dużego obciążenia i pozwala uzyskać dobre wyniki w przypadku systemu, w którym współczynnik zajętości  $\rho$  jest bliski 1. Oznacza to, że prawdopodobieństwo pustego systemu jest bliskie 0.

Ograniczenie to jest usuwane za pomocą innego warunku granicznego  $x = 0$ , czyli bariery z natychmiastowymi elementarnymi powrotami, które bywają też nazywane skokami [60]. Gdy proces dyfuzji osiągnie  $x = 0$ , pozostaje tam przez czas, który jest równy okresowi bezczynności, czyli okresowi z pustą kolejką bez przychodzących klientów, a następnie powraca do  $x = 1$ , co odpowiada przybyciu jednego pakietu do systemu.

Czas, w którym proces jest w stanie  $x = 0$ , oznacza, że system jest w stanie bezczynności. Równanie dyfuzji przybiera formę [60]:

$$\frac{\partial f(x, t; x_0)}{\partial t} = \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x} + \lambda p_0(t) \delta(x - 1), \quad (4.19)$$

$$\frac{dp_0(t)}{dt} = \lim_{x \rightarrow 0} \left[ \frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0) \right] - \lambda p_0(t), \quad (4.20)$$

gdzie  $p_0(t) = P[X(t) = 0]$ .

Wyrażenie  $\lambda p_0(t)\delta(x - 1)$  pozwala uzyskać gęstość prawdopodobieństwa rozpoczęcia procesu w punkcie  $x = 1$  w chwili  $t$ , z powodu skoku z bariery. Drugie równanie pozwala wyrazić  $p_0(t)$  za pomocą  $\lim_{x \rightarrow 0} [\frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0)]$ , co oznacza prawdopodobieństwo przepływu strumieni wchodzących do bariery, natomiast  $\lambda p_0(t)$  reprezentuje prawdopodobieństwo strumieni wychodzących z bariery [12].

Pewnym sposobem uzyskania funkcji  $f(x, t; x_0)$  procesu ze skokami z bariery, jest wyrażenie jej za pomocą innej funkcji gęstości  $\phi(x, t; x_0)$  dla procesu dyfuzji z barierą absorbującą w  $x = 0$  [66]. Proces ten rozpoczyna się w  $t = 0$  od  $x = x_0$  i kończy się, gdy osiągnie barierę. Taka funkcja gęstości prawdopodobieństwa jest łatwiejsza do wyznaczenia i obliczenia [67]:

$$\phi(x, t; x_0) = \frac{e^{\frac{\beta}{\alpha}(x-x_0) - \frac{\beta^2}{2\alpha}t}}{\sqrt{2\pi\alpha t}} \left[ e^{-\frac{(x-x_0)^2}{2\alpha t}} - e^{-\frac{(x+x_0)^2}{2\alpha t}} \right] \quad (4.21)$$

Natomiast funkcja gęstości czasu pierwszego przejścia z  $x = x_0$  do  $x = 0$  wyrażona jest [66]:

$$\gamma_{x_0,0}(t) = \lim_{x \rightarrow 0} \left[ \frac{\alpha}{2} \frac{\partial}{\partial x} \phi(x, t; x_0) - \beta \phi(x, t; x_0) \right] = \frac{x_0}{\sqrt{2\pi\alpha t^3}} e^{-\frac{(\beta t + x_0)^2}{2\alpha t}} \quad (4.22)$$

Przyjmując założenie, że proces z barierą ze skokami zaczyna się w czasie  $t = 0$  w punkcie  $x > 0$  o gęstości  $\psi(x)$ , a gdy dotrze do bariery, pozostaje tam przez czas określony za pomocą funkcji gęstości  $l_0(x)$ . Następnie skacze on do punktu  $x = 1$  i po pewnym czasie ponownie wraca do  $x = 0$ , po czym skacze do  $x = 1$  itd. Suma strumieni  $\gamma_0(t)$  mas prawdopodobieństwa które wchodzą do bariery wyznacza się jako [66]:

$$\gamma_0(t) = p_0(0)\delta(t) + [1 - p_0(0)]\gamma_{\psi,0}(t) + \int_0^t g_1(\tau)\gamma_{1,0}(t - \tau)d\tau, \quad (4.23)$$

gdzie:

$$\begin{aligned} \gamma_{\psi,0}(t) &= \int_0^\infty \gamma_{\xi,0}(t)\psi(\xi)d\xi, \\ g_1(\tau) &= \int_0^\tau \gamma_0(t)l_0(\tau - t)dt. \end{aligned} \quad (4.24)$$

Funkcję gęstości procesu dyfuzji z natychmiastowymi powrotami przedstawia się jako [66]:

$$f(x, t; x_0) = \phi(x, t; \psi) + \int_0^t g_1(\tau) \phi(x, t - \tau; 1) d\tau. \quad (4.25)$$

Oznaczmy przez  $\bar{f}(s)$  transformatę Laplace'a funkcji  $f(x)$ . Dla równań (4.23) i (4.24) transformaty Laplace'a wyrażone są więc następująco [66]:

$$\bar{\gamma}_0(s) = p_0(0) + [1 - p_0(0)]\bar{\gamma}_{\psi,0}(s) + \bar{g}_1(s)\bar{\gamma}_{1,0}(s),$$

$$\bar{g}_1(s) = \bar{\gamma}_0(s)\bar{l}_0(s), \quad (4.26)$$

gdzie:

$$\bar{\gamma}_{x_0,0}(s) = e^{-x_0 \frac{\beta + A(s)}{\alpha}}, \quad A(s) = \sqrt{\beta^2 + 2\alpha s},$$

$$\bar{\gamma}_{\psi,0}(s) = \int_0^\infty \bar{\gamma}_{\xi,0}(s) \psi(\xi) d\xi, \quad (4.27)$$

a następnie:

$$\bar{g}_1(s) = [p_0(0) + [1 - p_0(0)]\bar{\gamma}_{\psi,0}(s)] \frac{\bar{l}_0(s)}{1 - \bar{l}_0(s)\bar{\gamma}_{1,0}(s)}. \quad (4.28)$$

Równanie (4.25) w postaci transformaty Laplace'a przyjmuje postać [66]:

$$\bar{f}(x, s; x_0) = \bar{\phi}(x, s; \psi) + \bar{g}_1(s)\bar{\phi}(x, s; 1), \quad (4.29)$$

gdzie:

$$\bar{\phi}(x, s; x_0) = \frac{e^{\beta(x_0 x_0)}}{A(s)} [e^{-|x-x_0| \frac{A(s)}{\alpha}} - e^{-|x+x_0| \frac{A(s)}{\alpha}}],$$

$$\bar{\phi}(x, s; \psi) = \int_0^\infty \bar{\phi}(x, s; \xi) \psi(\xi) d\xi. \quad (4.30)$$

Odwrotne transformaty tych funkcji można uzyskać w sposób numeryczny. W tym celu stosuje się algorytm Stehfesta [68].

Przedstawiony w niniejszym podrozdziale model G/G/1 zakłada stałe parametry modelu. W przypadku przyjęcia zmiennych parametrów, model ten wymaga podzielenia czasu na takie okresy, dla których parametry modelu są stałe. W tym przypadku wyniki uzyskane na koniec jednego okresu stosowane są jako warunek początkowy dla kolejnego okresu.

### 4.3.2 Model ograniczonej kolejki: G/G/1/N

W przypadku systemu G/G/1/N, liczba pakietów na stanowisku ograniczona jest do  $N$ . Oznacza to, że model ten potrzebuje drugiej bariery w  $x = N$ . Proces dyfuzji dla modelu G/G/1/N ograniczony jest więc przez dwie bariery. Pierwsza z nich umiejscowiona w  $x = 0$  oraz druga w  $x = N$ . Zachowanie procesu w pierwszej barierze opisano w podrozdziale 4.3.1. W przypadku, gdy proces znajdzie się w drugiej barierze, czyli w  $x = N$ , pozostaje w niej tak długo jak kolejka jest pełna, a następnie przeskakuje do punktu  $x = N - 1$ . Funkcja gęstości takiego procesu  $f(x, t; x_0)$  opisana jest podobnie jak dla modelu z nieskończoną kolejką.

W pierwszym kroku, metoda określa gęstość  $\phi(x, t; x_0)$  procesu dyfuzji z dwoma barierami absorbującymi w  $x = 0$  i  $x = N$ , rozpoczynając w  $t = 0$  oraz  $x = x_0$  [67]:

$$\begin{aligned} \phi(x, t; x_0) = & \frac{1}{\sqrt{2\pi\alpha t}} \sum_{n=-\infty}^{\infty} \left( \exp\left[\frac{\beta x'_n}{\alpha} - \frac{(x - x_0 - x'_n - \beta t)^2}{2\alpha t}\right] \right. \\ & \left. - \exp\left[\frac{\beta x''_n}{\alpha} - \frac{(x - x_0 - x''_n - \beta t)^2}{2\alpha t}\right] \right), \end{aligned} \quad (4.31)$$

gdzie  $x'_n = 2nN$ ,  $x''_n = -2x_0 - x'_n$ .

Warunek początkowy zdefiniowany jest przez funkcję  $\psi(x)$ ,  $x \in (0, N)$ ,  $\lim_{x \rightarrow 0} \psi(x) = \lim_{x \rightarrow N} \psi(x) = 0$ , wtedy funkcja gęstości prawdopodobieństwa przyjmuje następującą formę:

$$\phi(x, t; \psi) = \int_0^N \phi(x, t; \xi) \psi(\xi) d\xi. \quad (4.32)$$

Gęstość  $f(x, t; \psi)$  procesu dyfuzji z powrotami z obu barier wyrazić można za pomocą funkcji  $\varphi(x, t; \psi)_1$ , reprezentującej wpływ warunków początkowych  $\psi$  oraz zbiór funkcji  $\varphi(x, t - \tau; 1)$ ,  $\varphi(x, t - \tau; N - 1)$ , rozpoczętych po skoku z bariery w czasie  $\tau < t$  w punktach  $x = 1$  oraz  $x = N - 1$  z intensywnością  $g_1(\tau)$  i  $g_{N-1}(\tau)$  [66]:

$$f(x, t; \psi) = \varphi(x, t; \psi) + \int_0^t g_1(\tau) \varphi(x, t - \tau; 1) d\tau + \int_0^t g_{N-1}(\tau) \varphi(x, t - \tau; N - 1) d\tau. \quad (4.33)$$

Funkcje  $g_1(t)$ ,  $g_N(t)$  można opisać za pomocą gęstości prawdopodobieństwa  $\gamma_0(t)$  i  $\gamma_N(t)$  [69]:

$$\begin{aligned} g_1(\tau) &= \int_0^\tau \gamma_0(t) l_0(\tau - t) dt, \\ g_{N-1}(\tau) &= \int_0^\tau \gamma_N(t) l_N(\tau - t) dt, \end{aligned} \quad (4.34)$$

gdzie  $l_0(x)$ ,  $l_N(x)$  to funkcje gęstości rozkładu czasu, w którym proces pozostaje w  $x = 0$  oraz  $x = N$ . Rozkłady te mogą być ogólne i nie muszą być ograniczone do rozkładów wykładniczych. Gęstości funkcji prawdopodobieństwa  $\gamma_0(t)$ ,  $\gamma_N(t)$  dla procesów, które znajdują się w barierze w  $x = 0$ , lub  $x = N$  w czasie  $t$  to [66]:

$$\begin{aligned} \gamma_0(t) &= p_0(0)\delta(t) + [1 - p_0(0) - p_N(0)]\gamma_{\psi,0}(t) \\ &+ \int_0^t g_1(\tau)\gamma_{1,0}(t - \tau)d\tau + \int_0^t g_{N-1}(\tau)\gamma_{N-1,0}(t - \tau)d\tau, \\ \gamma_N(t) &= p_N(0)\delta(t) + [1 - p_0(0) - p_N(0)]\gamma_{\psi,N}(t) \\ &+ \int_0^t g_1(\tau)\gamma_{1,N}(t - \tau)d\tau + \int_0^t g_{N-1}(\tau)\gamma_{N-1,N}(t - \tau)d\tau, \end{aligned} \quad (4.35)$$

gdzie  $\gamma_{1,0}(t)$ ,  $\gamma_{1,N}(t)$ ,  $\gamma_{N-1,0}(t)$ ,  $\gamma_{N-1,N}(t)$  są funkcjami gęstości czasu przejścia pomiędzy odpowiednimi punktami, wskazanymi w dolnym indeksie [66]:

$$\gamma_{1,0}(t) = \lim_{x \rightarrow 0} \left[ \frac{\alpha}{2} \frac{\partial \phi(x, t; 1)}{\partial x} - \beta \phi(x, t; 1) \right]. \quad (4.36)$$

Funkcje  $\gamma_{\psi,0}(t)$ ,  $\gamma_{\psi,N}(t)$  wyrażają gęstości prawdopodobieństwa, że procesy rozpoczęte w  $t = 0$ , w punkcie  $x = \xi$  z intensywnością  $\psi(\xi)$ , zakończą się w chwili  $t$ , znajdując się odpowiednio w barierze  $x = 0$  lub  $x = N$ .

Prawdopodobieństwa, że w chwili  $t$  proces ma wartość  $x = 0$  lub  $x = N$  wynoszą:

$$\begin{aligned} p_0(t) &= \int_0^t [\gamma_0(\tau) - g_1(\tau)] d\tau, \\ p_N(t) &= \int_0^t [\gamma_N(\tau) - g_{N-1}(\tau)] d\tau. \end{aligned} \quad (4.37)$$

Podobnie jak dla poprzedniego modelu, oryginały tak zdefiniowanych transformat Laplace'a dla funkcji  $\bar{f}(x, s; \psi)$  obliczane są numerycznie.

W przypadku zmiany współczynników dyfuzji konieczne jest odpowiednie dobranie przedziałów czasowych, w których parametry będą stałe, a rozwiązanie końca jednego przedziału wyznacza warunki początkowe dla kolejnego. Rozwiązanie to zmierza do  $t \rightarrow \infty$ . Dlatego też modele wykorzystujące dyfuzję aproksymacyjną mogą być również dotyczyć stanów ustalonych. Podejście takie stosowane jest między innymi w procesie oceny i ewaluacji mechanizmów aktywnego zarządzania kolejką w niniejszej rozprawie, a jego poprawność weryfikowana jest z wykorzystaniem metod symulacyjnych [70].

## 4.4 Model TCP/AQM aproksymacji dyfuzyjnej

W niniejszej sekcji przedstawiono dwa sposoby modelowania transmisji z wykorzystaniem protokołu TCP. Pierwszy sposób, opisany w podrozdziale 4.4.1, oparty jest na modelu, który wykorzystuje wyłącznie aproksymację dyfuzyjną.

Drugim sposobem zaproponowanym w ramach niniejszej rozprawy jest model łączący symulację zdarzeń dyskretnych z aproksymacją dyfuzyjną. Model ten szerzej opisano w podrozdziale 4.4.2. Cechuje się on zaletami aproksymacji dyfuzyjnej, a z drugiej strony zastosowanie metody symulacyjnej pozwala zwiększyć dokładność części dyfuzyjnej.

### 4.4.1 Czysty model aproksymacji dyfuzyjnej

Pierwszym wykorzystanym sposobem do modelowania transmisji dla protokołu TCP jest model aproksymacji dyfuzyjnej. W tym przypadku otrzymana wartość według poniższego wzoru (4.38) służy do obliczenia prawdopodobieństwa odrzucenia pakietów, a także do określenia rozmiaru okna przeciążenia w przypadku transmisji z pojedynczym strumieniem TCP:

$$\lambda = \frac{W\mu}{q} \quad (4.38)$$

gdzie za pomocą  $\lambda$  wyrażono poziom natężenia ruchu,  $W$  to rozmiar okna przeciążenia,  $1/\mu$  to średni czas obsługi, natomiast  $q/\mu$  oznacza opóźnienie routera.

Rozkład kolejki obliczony jest w czasie  $t_i$ , będącym końcem interwału  $i$ -tego. Wartość ta wykorzystywana jest do obliczenia średniej długości kolejki  $E[q_i]$ , a także do określenia prawdopodobieństwa odrzucenia pakietu  $p_i$ . Następnie wynik ten służy do zdefiniowania nowej wartości parametru  $\lambda$ :  $\lambda_{i+1} = \lambda_i + \Delta\lambda_i$ ,



istotnego z punktu widzenia mechanizmu aktywnego zarządzania kolejką, gdzie:

$$\Delta\lambda_i = \frac{\mu_i}{E[q_i]} - \frac{\lambda_i^2}{2} \frac{E[q_i]}{\mu} p_i. \quad (4.39)$$

Obliczenia te są powtarzane w czasie  $t_{i+1} = t_i + 1/\lambda_i$  dla nowych wartości parametru  $\lambda$ . W celu stworzenia przytoczonych modeli przyjęto założenia, zgodnie z którymi  $i - ty$  strumień może w dowolnej chwili rozpocząć lub zakończyć transmisję. Zmiana intensywności źródła  $\Delta\lambda$  w momencie  $t$  wpływa natomiast wprost na poziom zajętości kolejki.

#### 4.4.2 Model łączący symulację zdarzeń dyskretnych z aproksymacją dyfuzyjną

Drugi sposób modelowania transmisji TCP, który zaproponowano w ramach niniejszej rozprawy, łączy symulację zdarzeń dyskretnych z aproksymacją dyfuzyjną. W modelu tym, w zależności od obliczonej wartości średniej zajętości kolejki, część symulacyjna decyduje o odrzuceniu bądź przyjęciu pakietu. W przypadku zaimplementowania mechanizmu aktywnego zarządzania kolejką AQM decyzja o odrzuceniu pakietów jest losowa i zależy wprost od odpowiedzi mechanizmu, tj. od wartości funkcji prawdopodobieństwa odrzucenia pakietu.

Decyzja o odrzuceniu pakietu skutkuje zmniejszeniem o połowę wartości  $\lambda$ , odpowiadającej za intensywność napływania pakietów ze źródła. W przypadku braku strat pakietów, intensywność źródła wzrasta liniowo. Mechanizm ten postrzegać można również jako badanie natężenia ruchu TCP/IP w modelu pętli zamkniętej (ang. closed-loop) [1].

Przyjęty sposób zmiany intensywności źródła pakietów wyrazić można w następujący sposób:

$$\lambda = \begin{cases} \lambda + \zeta & \text{gdy } AQM \text{ decyduje o braku strat} \\ \frac{\lambda}{2} & \text{gdy } AQM \text{ decyduje o odrzuceniu pakietu} \end{cases} \quad (4.40)$$

gdzie  $\zeta$  zakłada stały wzrost źródła natężenia ruchu.

### 4.5 Model dla wielu niezależnych strumieni TCP i UDP

W sekcji tej przedstawiono definicje oraz założenia pozwalające na oszacowanie średniego poziomu zajętości kolejki oraz przeprowadzenie oceny mechanizmu

aktywnego zarządzania kolejką AQM, w przypadku transmisji z wieloma strumieniami TCP/UDP.

Suma strumieni wejściowych  $j$ , którą opisano dokładniej w [71, 72], posiada rozkład normalny o wartości średniej:

$$\lambda_j t = \left[ \sum_{i=1}^M \lambda_i r_{ij} + \lambda_{0j} \right] t \quad (4.41)$$

oraz wariancji:

$$\sigma_{Aj}^2 t = \left[ \sum_{i=1}^M C_{Dij}^2 \lambda_i r_{ij} + C_{0j}^2 \lambda_{0j} \right] t \quad (4.42)$$

skąd oblicza się:

$$C_{Aj}^2 = \frac{1}{\lambda_j} \sum_{i=1}^M r_{ij} \lambda_i [(C_{Di}^2 - 1) r_{ij} + 1] + \frac{C_{0j}^2 \lambda_{0j}}{\lambda_j}. \quad (4.43)$$

Parametry  $\lambda_{0j}$ ,  $C_{0j}^2$  oznaczają natomiast strumienie pochodzące od klientów zewnętrznych.

Wraz z upływem czasu  $t_{i+1} = t_i + 1/\lambda_i$ , powtarzane są obliczenia dla nowej wartości  $\lambda$ . W modelu przedstawionym w niniejszej pracy, zakłada się, że  $i - th$  strumień może rozpocząć lub zakończyć transmisję w dowolnym momencie. Zmiana intensywności źródła  $\Delta\lambda$  w czasie  $t$ , wpływa na czas potrzebny do wysłania pakietów oraz na długość kolejki.

Rozkład normalny liczby pakietów  $k$ -tego przychodzącego strumienia jest w przybliżeniu równy  $\lambda^{(k)} t$ , z wariancją spełniającą równanie  $\lambda^{(k)3} \sigma_A^{(k)2} t = \lambda^{(k)} C_A^{(k)2} t$ . Liczba pakietów ze wszystkich strumieni, które dotarły w tym czasie do stanowiska obsługi również ma rozkład normalny, wyrażony za pomocą średniej wartości  $\lambda t = \sum_{k=1}^K \lambda^{(k)} t$  oraz wariancji spełniającej równanie  $\lambda C_A^2 t = \sum_{k=1}^K \lambda^{(k)} C_A^{(k)2} t$ . Tak więc parametry wszystkich strumieni wyrażone są za pomocą:

$$\lambda(t) = \sum_{k=1}^K \lambda^{(k)}(t), \quad C_A^2 = \sum_{k=1}^K \frac{\lambda^{(k)}(t)}{\lambda(t)} C_A^{(k)2} \quad (4.44)$$

gdzie  $\frac{\lambda^{(k)}(t)}{\lambda(t)}$  jest prawdopodobieństwem, że dany pakiet należy do strumienia  $k$ , co pozwala wyznaczyć parametry czasu obsługi:

$$\frac{1}{\mu} = \sum_{k=1}^K \frac{\lambda^{(k)}}{\lambda} \frac{1}{\mu^{(k)}}, \quad C_B^2 = \mu^2 \sum_{k=1}^K \left[ \frac{\lambda^{(k)}}{\lambda} \frac{1}{\mu^{(k)^2} (C_B^{(k)^2} + 1)} \right] - 1, \quad (4.45)$$

a następnie parametry równania dyfuzji  $\alpha, \beta$ :

$$\beta(t) = \lambda(t) + \mu, \quad \alpha(t) = \lambda(t)C_A^2 + \mu C_B^2$$

Rozkład  $p(n) \approx f(n)$  określa liczbę pakietów wszystkich strumieni w kolejce, natomiast prawdopodobieństwo zdarzenia, że w kolejce znajdują się pakiety  $v$  należące do strumienia  $k$ -tego wynosi:

$$p^{(k)}(v) = \sum_{n=v}^{\infty} [p(n) \binom{n}{v} \left(\frac{\lambda^{(k)}}{\lambda}\right)^v \left(1 - \frac{\lambda^{(k)}}{\lambda}\right)^{n-v}], \quad (4.46)$$

gdzie  $k = 1, \dots, K$ .

W modelu, w którym rozważany jest przypadek dwóch rodzajów strumieni wejściowych, strumień UDP to strumień CBR o ustalonej liczbie pakietów wysyłanych w jednostce czasu, gdy wartość  $\lambda_{UDP}(t)$  jest stała, natomiast w przypadku strumienia TCP intensywność wejściowa zmienia się zgodnie z algorytmem kontroli przeciążenia TCP/AQM, wtedy wartość  $\lambda_{TCP}(t)$  kontrolowana jest przez mechanizm AQM. W przypadku modelu dyfuzji, wartość natężenia całkowitego strumienia wejściowego równa jest sumie natężeń jego wszystkich składników.

W przypadku dwóch strumieni wejściowych, jednego strumienia TCP oraz jednego UDP, wartość średnią oraz wariancję rozkładu wejściowego kolejki wyraża się za pomocą wzorów:

$$\begin{aligned} \beta(t) &= \lambda_{TCP}(t) + \lambda_{UDP}(t) + \mu & \alpha(t) &= [\lambda_{TCP}(t) + \lambda_{UDP}(t)] \\ & \left[ \frac{\sigma_{ATCP}^2(t) \lambda_{TCP}^3(t) + \sigma_{AUDP}^2(t) \lambda_{UDP}^3(t)}{\lambda_{TCP}(t) + \lambda_{UDP}(t)} \right] + \mu C_B^2 \\ \alpha(t) &= \sigma_{ATCP}^2(t) \lambda_{TCP}^3(t) + \sigma_{AUDP}^2(t) \lambda_{UDP}^3(t) + \mu C_B^2 \end{aligned} \quad (4.47)$$

Mechanizm TCP/AQM oparty na modelu aproksymacji dyfuzyjnej działa następująco: aproksymacja dyfuzyjna pozwala wyznaczyć rozkład kolejki routera w czasie  $t$ . Natomiast za pomocą średniej wartości zajętości kolejki modyfikowane jest prawdopodobieństwo odrzucenia pakietu. Prawdopodobieństwo to wpływa na intensywność  $l$  strumienia wejściowego, ponieważ okno przeciążenia zwiększa się o jeden w przypadku bezstratnej transmisji lub jest zmniejszane o połowę w przypadku utraty pakietów.

## 4.6 Model sieci wielu stanowisk obsługi

W przypadku modelu sieci wielu stanowisk obsługi, suma pakietów przychodzących na stanowisko  $j$  w czasie  $\Delta t$  ma rozkład normalny ze średnią:

$$\lambda_j \Delta t = \left[ \sum_{i=1}^M \lambda_i r_{ij} + \lambda_{0j} \right] \Delta t \quad (4.48)$$

gdzie  $r_{ij}$  jest prawdopodobieństwem routingu między stanowiskami  $i$  oraz  $j$ . Liczba połączonych stanowisk wyrażona jest za pomocą  $M$ , natomiast wariancja przedstawiona jest wzorem:

$$\sigma_{Aj}^2 \Delta t = \left[ \sum_{i=1}^M C_{Di}^2 \lambda_i r_{ij} + C_{0j}^2 \lambda_{0j} \right] \Delta t. \quad (4.49)$$

gdzie  $C_{Di}^2$  jest kwadratem współczynnika zmienności czasów przybycia pakietów do dowolnego stanowiska  $i$ .

Za pomocą przytoczonych równań, dla każdego stanowiska  $i$ , wyznaczyć można układ, który pozwala obliczyć  $\lambda_i$  oraz kwadrat współczynnika zmienności między czasem przybycia pakietów do dowolnego stanowiska  $j = 1, \dots, M$ :

$$C_{Aj}^2 = \frac{1}{\lambda_j} \sum_{i=1}^M r_{ij} \lambda_i [(C_{Di}^2 - 1)r_{ij} + 1] + \frac{C_{0j}^2 \lambda_{0j}}{\lambda_j}, \quad (4.50)$$

skąd wyznaczyć można parametry modelu dyfuzyjnego. Parametry  $\lambda_{0j}$ ,  $C_{0j}^2$  odnoszą się natomiast do strumieni pochodzących ze stanowiska  $j$ .

# Rozdział 5

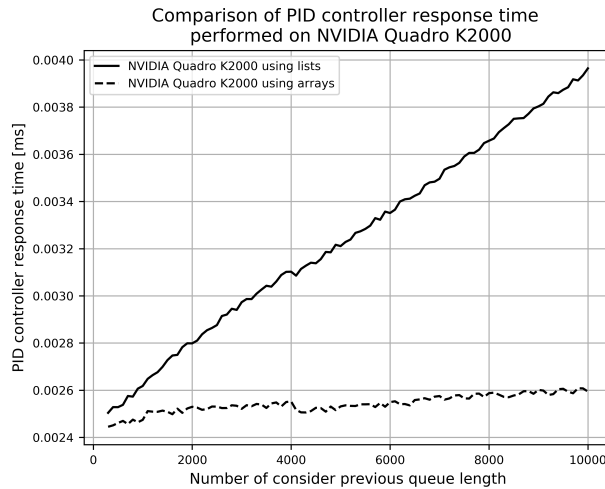
## Opis przedstawionych publikacji

W rozdziale tym zawarto zbiór pięciu publikacji naukowych, które przedstawiają szczegółowy opis zaproponowanych modeli oraz przeprowadzonych eksperymentów badawczych zrealizowanych w ramach niniejszej dysertacji.

### 5.1 Publikacja 1: GPU accelerated non-integer order $PI^\alpha D^\beta$ controller used as AQM mechanism

Celem pracy "GPU accelerated non-integer order  $PI^\alpha D^\beta$  controller used as AQM mechanism" było przyspieszenie czasu obliczeń kontrolera  $PI^\alpha D^\beta$  z wykorzystaniem struktur GPU oraz technologii CUDA. Zgodnie z moją najlepszą wiedzą, w chwili prowadzenia niniejszych badań nie istniały analizy implementacji mechanizmów aktywnego zarządzania kolejką w oparciu o obliczenia przeprowadzane w strukturach GPU. W ramach prac zaimplementowano kontroler  $PI^\alpha D^\beta$  niecałkowitego rzędu z wykorzystaniem różnych platform sprzętowych. W trakcie prac badawczych skupiono się na analizie czasu potrzebnego na przesył danych do struktur GPU oraz wpływie liczby elementów historii zajętości kolejki na czas potrzebny na uzyskanie odpowiedzi. Wykorzystanie struktur GPU wiąże się również z dodatkowymi opóźnieniami związanymi z przesyłaniem danych między jednostką CPU, a procesorem graficznym GPU. W trakcie badań ustalono, że czas ten wynosi około 12% całkowitego czasu obliczeń. Ponadto dane przesyłane do struktur GPU muszą posiadać postać tablicy. W związku z tym przeprowadzenie jakiegokolwiek konwersji dodatkowo wpływa na czas odpowiedzi kontrolera. Na wykresie 5.1 przedstawiono kontroler  $PI^\alpha D^\beta$  zaimplementowany w strukturach GPU oraz wpływ zastosowania list oraz tablic, na czas odpowiedzi mechanizmu

AQM. Wyniki przeprowadzonych prac badawczych dowodzą, że wielkość prze-



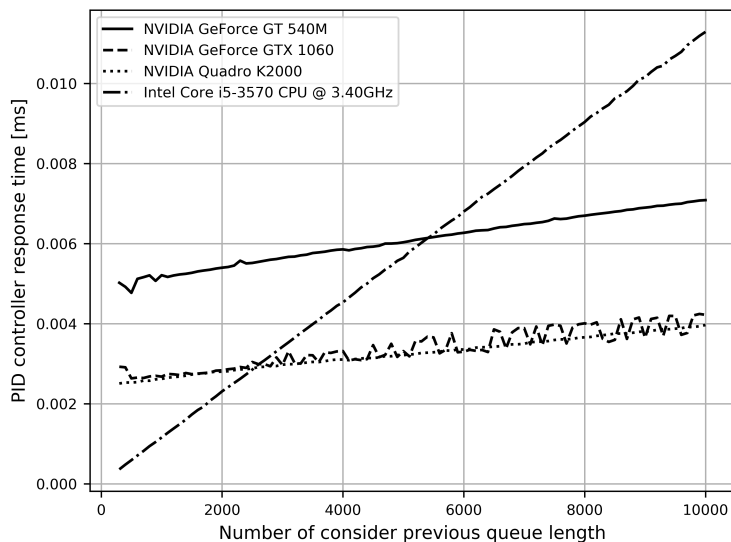
Rysunek 5.1: Porównanie czasu odpowiedzi kontrolera  $PI^\alpha D^\beta$  w przypadku implementacji w strukturach GPU, opartej o listy oraz tablice.

syłanych danych nie ma znaczącego wpływu w przypadku implementacji opartej o tablice.

Na wykresie 5.2 przedstawiono natomiast porównanie czasu odpowiedzi kontrolera niecałkowitego rzędu  $PI^\alpha D^\beta$ , w przypadku wykorzystania struktur GPU, CPU, a także zmiennej liczby elementów historii zajętości kolejki uwzględnianych w obliczeniach.

Uzyskane rezultaty wskazują, że czas odpowiedzi kontrolera  $PI^\alpha D^\beta$  w przypadku struktur GPU nie wzrasta tak gwałtownie wraz ze wzrostem liczby elementów historii zajętości kolejki, jak w przypadku obliczeń opartych jedynie o jednostkę procesora. Ponadto wyniki te dowodzą, że implementacja w strukturach GPU pozwala zoptymalizować pracę kontrolera, zwłaszcza wtedy, gdy rozmiar historii zajętości kolejki przekracza pewną określoną wartość dla danej jednostki sprzętowej. W przypadku niniejszych badań, liczba ta zbliżona była do 2000 elementów.

Rezultaty przeprowadzonych prac badawczych potwierdziły, że wykorzystanie implementacji kontrolera niecałkowitego rzędu  $PI^\alpha D^\beta$  w strukturach GPU pozwala skrócić czas odpowiedzi mechanizmu AQM. Zaprezentowane podejście pozwala uzyskać szczególną wydajność w przypadku analizy długich sekwencji historii zajętości kolejki. Jednak najistotniejszym wnioskiem jest potwierdzenie



Rysunek 5.2: Porównanie czasu odpowiedzi kontrolera  $PI^\alpha D^\beta$  z wykorzystaniem różnych struktur GPU, jednostki CPU oraz zmiennej liczby elementów historii zajętości kolejki.

nie możliwości zoptymalizowania czasu odpowiedzi kontrolera  $PI^\alpha D^\beta$  jako mechanizmu AQM, z którego wykorzystaniem wiąże się wykonywanie obliczeń o dużej złożoności. W przypadku braku stwierdzenia takiej możliwości, dalsze prace związane z próbą zastosowania kontrolera  $PI^\alpha D^\beta$  jako mechanizmu AQM w węzle komunikacyjnym praktycznie pozbawione zostałyby sensu. W tym przypadku ich praktyczne zastosowanie w rzeczywistej infrastrukturze sieciowej byłoby w zasadzie nieosiągalne.

Wkład autorski [60%] związany był z:

- współudziałem w wykonaniu przeglądu literatury,
- współudziałem w zaproponowaniu badań mających na celu przyspieszenie czasu obliczeń odpowiedzi kontrolera niecałkowitego rzędu  $PI^\alpha D^\beta$ , w oparciu o struktury GPU,
- zaimplementowaniem zaproponowanego modelu z wykorzystaniem struktur CPU oraz GPU,

- przeprowadzeniem badań eksperymentalnych,
- współdziałaniem w analizie uzyskanych wyników oraz sformułowaniem wniosków.

## 5.2 Publikacja 2: Combined diffusion approximation - simulation model of AQMs transient behaviour

W pracy "Combined diffusion approximation - simulation model of AQMs transient behaviour" przeprowadzono badania mające na celu wykorzystanie aproksymacji dyfuzyjnej do oceny mechanizmów aktywnego zarządzania kolejką AQM (ang. Active Queue Management). Zaletą zastosowanej metody dyfuzji jest zdolność do rozważenia zarówno cech pierwszorzędnych ruchu, tj. jego intensywności, ale również zmienności wyrażonej za pomocą wariancji. Inne znane metody umożliwiające ocenę mechanizmów aktywnego zarządzania kolejką, takie jak aproksymacja Fluid-Flow, umożliwiają jedynie określenie przybliżonego rozkładu średniej zajętości kolejki. Opublikowany w niniejszej pracy model łączy aproksymację dyfuzyjną z symulacją zdarzeń dyskretnych. Przewagą takiego rozwiązania jest uzyskanie opisu stanów nieustalonych dla dokładnych rozkładów czasu obsługi pakietów między kolejnymi zgłoszeniami do stanowiska, przy zastosowaniu kolejki z mechanizmem aktywnego zarządzania pakietami. Część symulacyjna modelu decyduje o stratach pakietów i modyfikuje natężenie przepływu pakietów, które wysyłane są przez nadajnik. Natomiast zastosowanie aproksymacji dyfuzyjnej pozwala na dokładne szacowanie dystrybucji pakietów.

Na podstawie tak zdefiniowanego modelu badawczego przeprowadzono ewaluację kontrolera RED oraz regulatora niecałkowitego rzędu  $PI^n$ , które zastosowane zostały jako mechanizmy aktywnego zarządzania kolejką. W fazie eksperymentalnej do oceny zaproponowanych mechanizmów brano pod uwagę takie parametry transmisji jak średnia długość kolejki oraz zmiany intensywności źródła, które wyrażono za pomocą  $\lambda$ .

Obliczenia dyfuzji przeprowadzono z krokiem  $\Delta t = \frac{1}{\lambda}$ . Na wyjściu modelu otrzymywany jest rozkład zajętości kolejki. Na tej podstawie część symulacyjna modelu decyduje o odrzuceniu, bądź też przyjęciu pakietu. Decyzja ta wpływa następnie na nową wartość intensywności źródła  $\lambda$ .



|   | $K_p$  | $K_i$  | $\eta$ | setpoint |
|---|--------|--------|--------|----------|
| 1 | 0.0001 | 0.0004 | -1.2   | 10       |
| 2 | 0.0001 | 0.0004 | -1.0   | 10       |
| 3 | 0.0001 | 0.0014 | -0.8   | 10       |
| 4 | 0.0001 | 0.0040 | -0.4   | 10       |

Tablica 5.1: Współczynniki kontrolera niecałkowitego rzędu  $PI^\eta$

| AQM         | Diffusion | Simulation |
|-------------|-----------|------------|
| <i>RED</i>  | 18.067    | 17.106     |
| $PI^\eta$ 1 | 5.100     | 5.251      |
| $PI^\eta$ 2 | 6.122     | 6.302      |
| $PI^\eta$ 3 | 7.780     | 7.123      |
| $PI^\eta$ 4 | 10.431    | 10.075     |

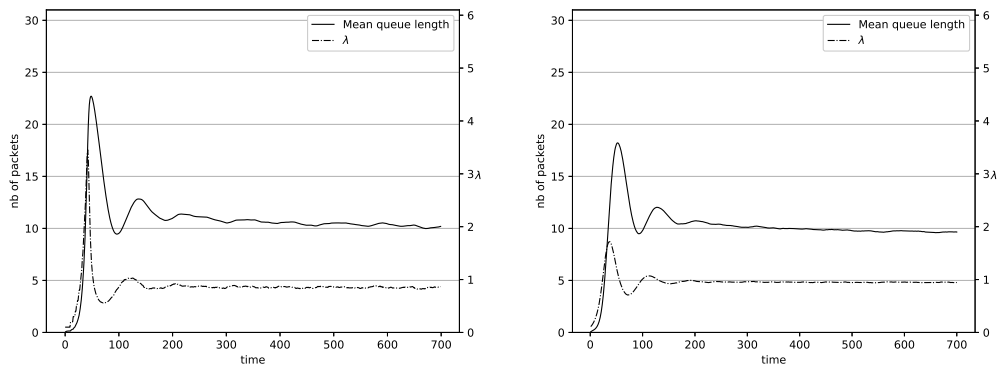
Tablica 5.2: Uzyskane średnie rezultaty zajętości kolejki dla wybranych mechanizmów aktywnego zarządzania kolejką.

Parametry kontrolera niecałkowitego rzędu  $PI^\eta$  dobrano na podstawie rezultatów wcześniejszych prac badawczych [42, 73, 74]. Wartości tych współczynników zestawiono w tabeli 5.1.

W przypadku gdy  $\eta = -1$ , wtedy oznacza to zastosowanie klasycznego kontrolera PI. Maksymalną długość kolejki w badaniach ustalono jako 30, natomiast wartość oczekiwana dla kontrolera  $PI^\eta$  to 10. W przypadku kontrolera RED ustalono wartość  $Min_{th} = 10$  oraz  $Max_{th} = 20$ .

W przeprowadzonych badaniach porównano rezultaty otrzymane z wykorzystaniem zaproponowanej metody łączącej symulację z aproksymacją dyfuzyjną, w porównaniu do klasycznej metody symulacyjnej. W tabeli 5.2 zaprezentowano uzyskane średnie wartości zajętości kolejki. Natomiast szczegółowe rezultaty zajętości kolejki w stanie ustalonym dla kontrolera  $PI^\eta$  4 przedstawiono na wykresie 5.3.

Uzyskane wyniki potwierdzają wpływ parametrów mechanizmu na zachowanie bufora transmisji. Średnia długość kolejki wyraźnie maleje wraz ze wzrostem parametru niecałkowitego rzędu. Z kolei zaimplementowanie modelu opartego o aproksymację dyfuzyjną sprawia, że czas wykonania pojedynczego eksperymentu ulega znacznemu wydłużeniu. Jego wykorzystanie pozwala jednak uzyskać dokładniejsze wyniki oraz szczegółowe rezultaty zajętości kolejki w stanie ustalonym.



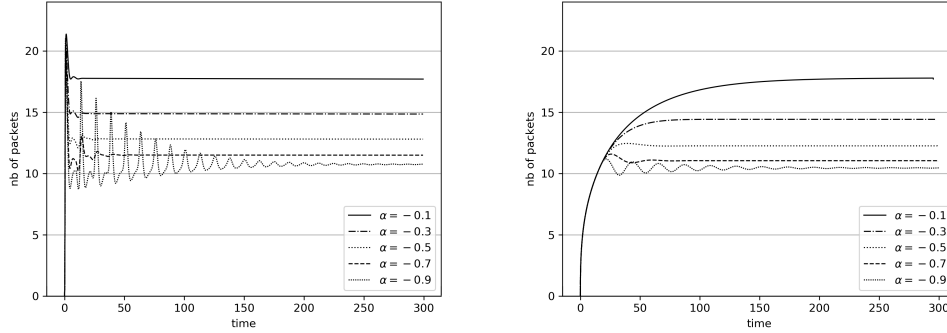
Rysunek 5.3: Średnia długość kolejki routera oraz intensywność źródła transmisji, model oparty o aproksymację dyfuzyjną (lewy wykres), klasyczną metodę symulacyjną (prawy wykres), kontroler  $PI^\eta$  4.

Wkład autorski [65%] związany był z:

- współudziałem w wykonaniu przeglądu literatury,
- zaproponowaniem modelu łączącego symulację zdarzeń dyskretnych z aproksymacją dyfuzyjną, który umożliwia ocenę mechanizmów aktywnego zarządzania kolejką,
- zaimplementowaniem modelu dyfuzyjnego oraz symulacyjnego,
- przeprowadzeniem badań eksperymentalnych,
- analizą uzyskanych wyników oraz zdefiniowaniem wniosków końcowych.

### 5.3 Publikacja 3: Diffusion approximation model of TCP NewReno congestion control mechanism

W artykule "Diffusion approximation model of TCP NewReno congestion control mechanism" zaproponowano model wykorzystujący dyfuzję aproksymacyjną do modelowania zachowania strumienia TCP. Koncepcja ta stanowi rozwinięcie badań związanych z połączeniem symulacji zdarzeń dyskretnych z aproksymacją dyfuzyjną do ewaluacji mechanizmów aktywnego zarządzania kolejką.



Rysunek 5.4: Średnia zajętość kolejki, aproksymacja dyfuzyjna (lewy wykres), aproksymacja Fluid-Flow (prawy wykres), kontroler  $PI^\alpha$ :  $K_P = 0,0001$ ,  $K_I = 0,0004$ , parametr  $\alpha$  w zakresie  $[-0, 1, -0, 9]$

W ramach niniejszej pracy przeprowadzono badania, które pozwoliły określić poprawność zaproponowanego modelu dyfuzyjnego, względem klasycznej metody analitycznej, opartej o aproksymację Fluid-Flow.

W przeprowadzonych eksperymentach numerycznych zastosowano kolejkę  $G/G/1/L$ , gdzie wielkość bufora to  $L = 30$ . Definicje oraz założenia, które umożliwiły oszacowanie średniego poziomu zajętości kolejki, przedstawione zostały we wzorach 4.38 oraz 4.39, w rozdziale 4.6 niniejszej rozprawy.

Zastosowanie zaproponowanego modelu dyfuzyjnego pozwoliło na uzyskanie pełnego rozkładu kolejki, a nie tylko wartości średnich. Rozkład kolejki  $f(n, t, n_0)$  pozwala natomiast obliczyć funkcję gęstości  $r(t, x)$  opóźnień występujących w transmisji:

$$r(t, x) = \sum_{n=0}^L f(n, r; n_0) b(x)^{(n+1)*} \quad (5.1)$$

gdzie  $b(x)$  jest funkcją gęstości prawdopodobieństwa czasu obsługi pakietu, co z kolei pozwala wyznaczyć maksymalne różnice opóźnień między pakietami w łączy transmisyjnym, a więc wariancję ruchu (ang. jitter).

Na wykresie 5.4 przedstawiono uzyskane rezultaty zajętości kolejki w przypadku zastosowania kontrolera niecałkowitego rzędu  $PI^\alpha$  z wykorzystaniem zaproponowanego modelu opartego o aproksymację dyfuzyjną (lewy wykres 5.4) oraz aproksymację Fluid-Flow (prawy wykres 5.4).

Uzyskane rezultaty potwierdziły poprawność zaproponowanego modelu.

Otrzymane średnie wartości zajętości kolejki były bardzo zbliżone do rezultatów przy wykorzystaniu klasycznej metody opartej o aproksymację Fluid-Flow. Wyniki te wykazują silny wpływ parametru  $\alpha$  kontrolera niecałkowitego rzędu PI, na poziom zajętości kolejki. Wraz z jego spadkiem, średnie odnotowane wartości zajętości bufora maleją od 17.764 (gdy  $\alpha = -0.1$ ) do 10.448 (dla  $\alpha = -0.9$ ).

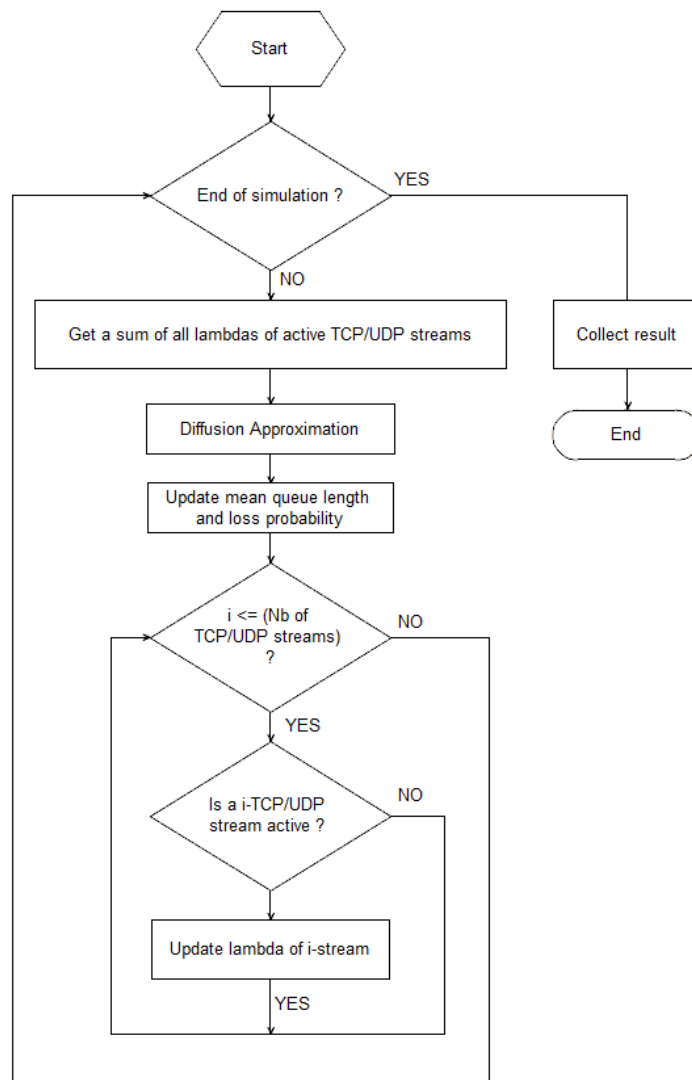
Wkład autorski [60%] związany był z:

- współudziałem w wykonaniu przeglądu literatury,
- zaproponowaniem modelu łączącego symulację zdarzeń dyskretnych z aproksymacją dyfuzyjną, który umożliwia ocenę mechanizmów aktywnego zarządzania kolejką,
- zaimplementowaniem modelu dyfuzyjnego oraz modelu opartego o analityczną metodę Fluid-Flow,
- przeprowadzeniem badań eksperymentalnych,
- analizą uzyskanych wyników oraz współudziałem w sformułowaniu wniosków.

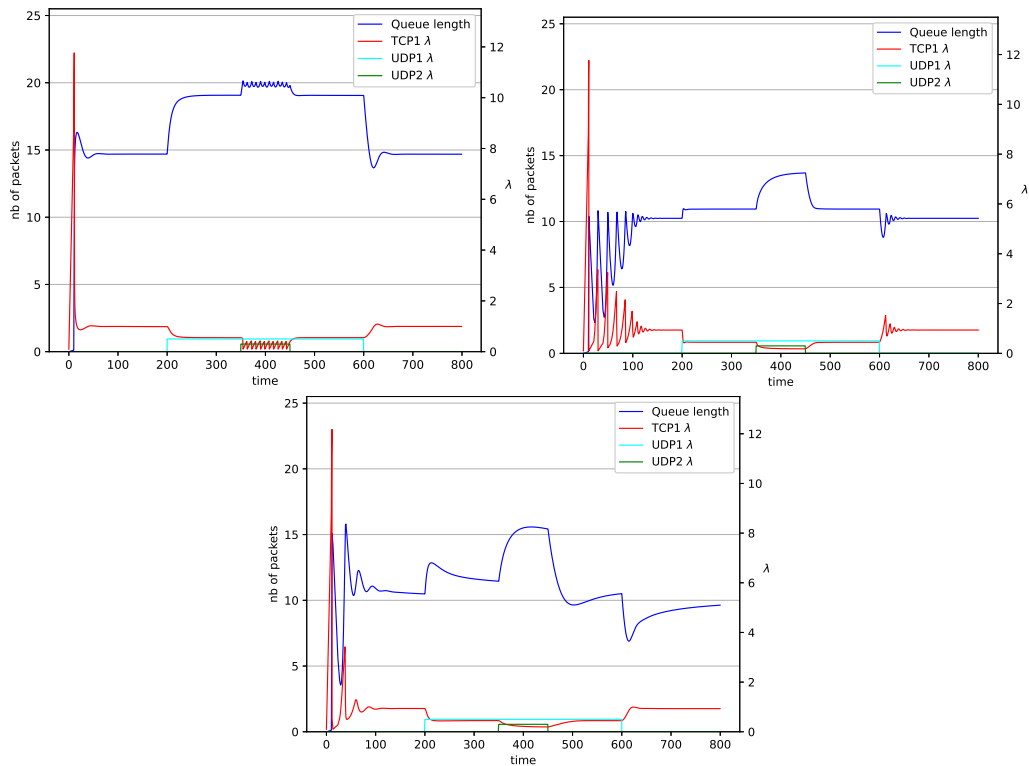
## 5.4 Publikacja 4: Diffusion Model of a Non-Integer Order $PI^\gamma$ Controller with TCP/UDP Streams

Publikacja "Diffusion Model of a Non-Integer Order  $PI^\gamma$  Controller with TCP/UDP Streams" stanowi kontynuację wcześniejszych prac związanych z zaproponowaniem modelu łączącego symulację zdarzeń dyskretnych z aproksymacją dyfuzyjną, który służy do oceny mechanizmów aktywnego zarządzania kolejką, w tym kontrolera  $PI^\gamma$  niecałkowitego rzędu. W ramach niniejszej pracy przeprowadzono badania mające na celu ocenę wydajności transmisji przy zastosowaniu większej liczby strumieni TCP/UDP. Uzyskane rezultaty pozwoliły ocenić dynamikę transmisji, kiedy różne źródła TCP/UDP rozpoczynają lub kończą transmisję. Wykazały także zalety wynikające z zastosowania kontrolera PI niecałkowitego rzędu, względem tradycyjnego mechanizmu AQM, w tym zmniejszenia fluktuacji rozkładu zajętości kolejki. W tym artykule rozszerzona została funkcjonalność mechanizmu wykorzystującego aproksymację dyfuzyjną do modelowania zachowania strumienia TCP, o możliwość przeanalizowania większej ilości strumieni TCP/UDP. Z drugiej strony zaproponowane podejście umożliwia

zupełnie nowe spojrzenie na współpracę protokołu TCP/UDP z mechanizmami AQM. Model ten pozwala na rozpoczęcie i zakończenie transmisji TCP/UDP w dowolnym momencie. Umożliwia to obserwację parametrów transmisji oraz ich wpływu na ewolucję okna przeciążenia TCP, podczas gdy różne źródła rozpoczynają lub kończą przesył danych. Schemat blokowy algorytmu wykorzystywanego w niniejszej pracy przedstawiono na rysunku 5.5.



Rysunek 5.5: Schemat blokowy algorytmu w zaproponowanym modelu.



Rysunek 5.6: Transmisja dla jednego strumienia TCP oraz dwóch strumieni UDP, kontroler RED (lewy wykres), kontroler NLRED (prawy wykres), kontroler  $PI^\gamma$  niecałkowitego rzędu (środkowy wykres).

Na wykresie 5.6 przedstawiono transmisję z jednym aktywnym strumieniem TCP oraz dwoma zmiennymi strumieniami UDP, w przypadku zastosowania kontrolera RED oraz  $PI^\gamma$  niecałkowitego rzędu. Pierwszy strumień UDP rozpoczyna transmisję w czasie  $t = 200$  i kończy w  $t = 600$ . Drugi strumień UDP jest aktywny między  $t = 350$  a  $t = 450$  i powoduje przeciążenie kolejki. W obu przypadkach, zarówno w przypadku mechanizmu RED, jak też dla kontrolera  $PI^\gamma$ , poziom zajętości kolejki dynamicznie wzrasta. Jednak w przypadku wykorzystania mechanizmu  $PI^\gamma$  zajętość kolejki powoli zaczyna się stabilizować, a następnie zmniejszać (środkowy wykres 5.6). Obserwacja taka nie zachodzi w przypadku mechanizmu RED oraz NLRED (lewy oraz prawy wykres 5.6), ponieważ nie są one w stanie efektywnie obsłużyć takiej ilości danych wejściowych. Ponadto w przypadku algorytmu RED wielkość kolejki oscyluje wokół punktu  $Max_{th}$ .

Przeprowadzone eksperymenty badawcze wykazały różnice w uzyskanych parametrach transmisji oraz poziomie zajętości bufora, w przypadku zastosowania różnych mechanizmów AQM. Uwidocznily również zalety wykorzystania kontrolera niecałkowitego rzędu  $PI^\gamma$  jako mechanizmu aktywnego zarządzania kolejką. Ponadto dodatkowym wnioskiem z fazy eksperymentalnej jest obserwacja potwierdzająca, że zmniejszenie wartości rzędu całkowania kontrolera  $PI^\gamma$  skutkuje zmniejszeniem fluktuacji długości kolejki, a przez to pośrednio wpływa na zmniejszenie zmienności natężenia ruchu sieciowego.

Wkład autorski [65%] związany był z:

- współudziałem w wykonaniu przeglądu literatury,
- zaproponowaniem modelu opartego o aproksymację dyfuzyjną, który pozwala ocenić wpływ zastosowania kontrolera  $PI^\gamma$  niecałkowitego rzędu, przy zastosowaniu większej liczby strumieni TCP/UDP,
- zaimplementowaniem modelu dyfuzyjnego oraz symulacyjnego,
- przeprowadzeniem badań eksperymentalnych,
- analizą uzyskanych wyników oraz zdefiniowaniem wniosków końcowych.

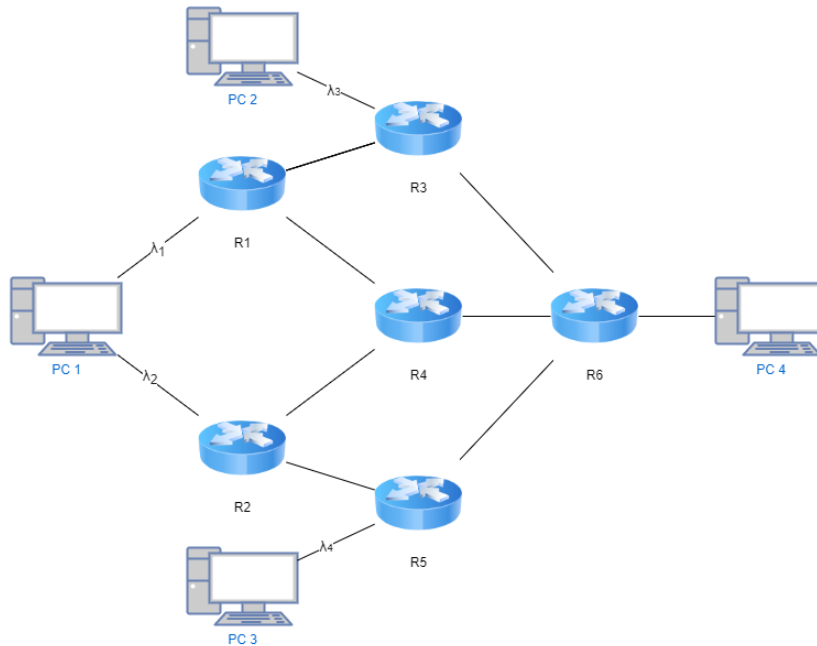
## 5.5 Publikacja 5: Approximation Models for the Evaluation of TCP/AQM Networks

Artykuł "Approximation Models for the Evaluation of TCP/AQM Networks" przedstawia sieciowy model wykorzystujący aproksymację dyfuzyjną do oceny współpracy mechanizmu okna przeciążenia TCP i jego kooperacji z mechanizmami aktywnego zarządzania kolejką. Model rozpatruje możliwość istnienia jednocześnie wielu strumieni TCP oraz UDP, a także stanowi wielowęzłowe rozszerzenie koncepcji przyjętej we wcześniejszej pracy "Diffusion Model of a Non-Integer Order  $PI^\gamma$  Controller with TCP/UDP Streams", przedstawionej w niniejszej rozprawie.

W fazie eksperymentalnej, oprócz kontrolera niecałkowitego rzędu  $PI^\gamma$ , wykorzystano również mechanizm RED oraz NLRED, stanowiące klasyczne rozwiązania z zakresu algorytmów aktywnego zarządzania kolejką.

Zaproponowany model zweryfikowano z wykorzystaniem aproksymacji Fluid-Flow. Model ten stworzony został w celu ewaluacji współpracy mechanizmu AQM z protokołem TCP. Zgodnie z wnioskami z prac wcześniejszych, zastosowanie modelu dyfuzyjnego pozwala oprócz intensywności źródła uwzględnić również jego zmienność, czyli wariancję. Jego zaimplementowanie umożliwia także rozpatrywanie bardziej złożonych modeli sieciowych.

Topologia sieci, którą zaimplementowano do zaplanowanych badań, przedstawiona została na rysunku 5.7. Rozważany system składał się więc z sześciu routerów, trzech stacji nadawczych oraz jednej stacji odbiorczej. Aby zweryfikować poprawność otrzymanych wyników, wykonano 100 tysięcy symulacji sieciowych, mających na celu uśrednienie otrzymanych wyników dla pojedynczego eksperymentu.

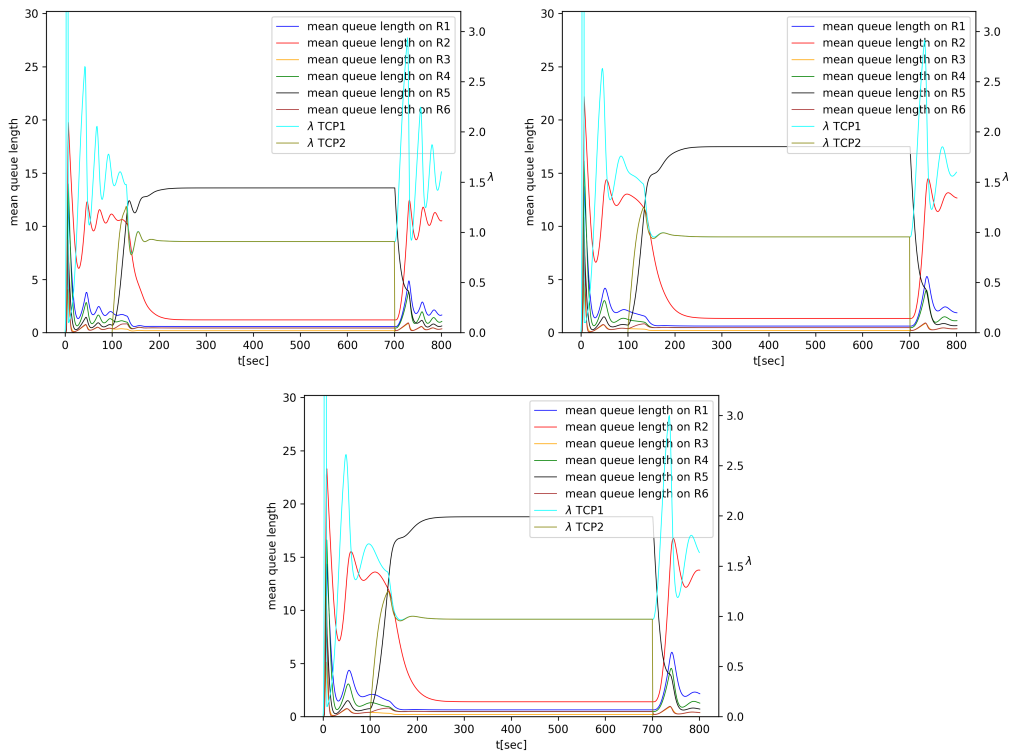


Rysunek 5.7: Schemat przyjętego modelu sieci składającej się z sześciu routerów, trzech stacji nadawczych oraz jednej stacji odbiorczej.

Wykresy przedstawione na rysunku 5.8 prezentują wyniki uzyskane w przypadku aktywnej transmisji z dwoma strumieniami TCP.

Strumień TCP2 rozpoczyna się w czasie  $t = 100$ , kończy w  $t = 700$  i transmitowany jest z wykorzystaniem routera R5 ( $\lambda_4$ ). Strumień TCP1 rozchodzi się





Rysunek 5.8: Średnia zajętość kolejki dla transmisji z dwóch strumieniami TCP, kontroler NLRED (lewy wykres), kontroler RED (prawy wykres), kontroler  $PI^\gamma$  (środkowy wykres).

natomiast na routerach R1 ( $\lambda_1$ ) oraz R2 ( $\lambda_2$ ). Analiza otrzymanych rezultatów wskazuje, że oba strumienie dążą do maksymalnego wykorzystania łącza. Jednakże największy wpływ zauważalny jest na węzłach bezpośrednio podłączonych do stacji nadawczych. Zajętość węzłów bardziej oddalonych od źródła ruchu jest w tym przypadku niewielka.

W ramach zaplanowanych prac zrealizowano również szczegółowe badania związane z analizą wydajności dla kontrolera  $PI^\gamma$  niecałkowitego rzędu, względem tradycyjnych mechanizmów RED oraz NLRED. Rozważano scenariusze dla jednego strumienia TCP, w których pomiędzy nadajnikiem, a odbiornikiem ustanowiono jeden, a następnie trzy routery pośredniczące. Kolejne fazy eksperymentów przeprowadzono z dodatkowymi strumieniami UDP. Jednak wszystkie uzyskane rezultaty wskazały, że w przypadku zastosowania kontrolera  $PI^\gamma$  niecałko-

witego rzędu jako mechanizmu AQM, zaobserwowano większe średnie zajętości kolejki oraz optymalne wykorzystanie łącza transmisyjnego. Rezultaty otrzymane z wykorzystaniem aproksymacji Fluid-Flow potwierdziły poprawność zaproponowanego modelu. Metoda ta jest tradycyjnym rozwiązaniem stosowanym do oceny współpracy mechanizmów aktywnego zarządzania kolejką z protokołem TCP.

Dzięki zastosowaniu modelu dyfuzyjnego możliwe było rozpatrywanie ruchu, który nie rozkłada się w sposób równomierny na wszystkich dostępnych węzłach w sieci. Pozwala to na rozpatrywanie bardziej złożonych modeli i topologii. Ponadto model dyfuzyjny dostarcza informacje o rozkładzie kolejek oraz o wariancji ruchu sieciowego, co z kolei pozwala na lepszą obserwację stanów przejściowych oraz dynamiki transmisji.

Wkład autorski [60%] związany był z:

- współudziałem w wykonaniu przeglądu literatury,
- zaproponowaniem modelu opartego o aproksymację dyfuzyjną, który pozwala ocenić mechanizmy aktywnego zarządzania kolejką,
- zaimplementowaniem wielowęzłowej implementacji modelu dyfuzyjnego,
- zaimplementowaniem metody aproksymacji Fluid-Flow, która umożliwiła zweryfikowanie zaproponowanego modelu,
- przeprowadzeniem eksperymentów badawczych,
- analizą uzyskanych wyników oraz sformułowaniem wniosków końcowych.

## **Rozdział 6**

### **GPU accelerated non-integer order $PI^\alpha D^\beta$ controller used as AQM mechanism**



# GPU Accelerated Non-integer Order $PI^\alpha D^\beta$ Controller Used as AQM Mechanism

Adam Domański<sup>1</sup>( ), Joanna Domańska<sup>2</sup>, Tadeusz Czachórski<sup>2</sup>,  
Jerzy Klamka<sup>2</sup>, Dariusz Marek<sup>1</sup>, and Jakub Szygula<sup>1</sup>

<sup>1</sup> Institute of Informatics, Silesian University of Technology,  
Akademicka 16, 44-100 Gliwice, Poland

{adomanski,dariusz.marek,jakub.szygula}@polsl.pl

<sup>2</sup> Institute of Theoretical and Applied Informatics, Polish Academy of Sciences,  
ul. Bałtycka 5, 44-100 Gliwice, Poland

{joanna,tadek,jerzy.klamka}@iitis.pl

**Abstract.** The goal of this article was to reduce the time of dropping packed decisions for Active Queue Mechanisms based on  $PI^\alpha D^\beta$ . This article introduces a new way of performing calculations in a non-integer order controller using CUDA technology.

**Keywords:** Active Queue Management · Non-integer order  $PI^\alpha D^\beta$   
CUDA · GPU · Packet loss · Network congestion control

## 1 Introduction

Nowadays, the congestion problems are common in communication networks. The network routers send packets via links with limited bandwidth. If the number of incoming packets exceeds the capacity of a link, it becomes congested and the buffers of routers, that store packets, may overflow.

To solve this kind of problem in TCP/IP networks, the IETF organization (Internet Engineering Task Force), which develops and promotes Internet standards, recommends the use of Active Queue Management (AQM) mechanisms in IP routers. These mechanisms proactively drop packets, even if there is a place to store them, in order to announce that the router queue is increasing and there is a risk of congestion. The probability of packet rejection increases with the level of congestion. Due to the fact that packets are dropped randomly, only some users are notified and the phenomenon of global synchronization is therefore avoided. The mechanisms of active queue management enhance the efficiency of transmissions and cooperate with the congestion window mechanism of TCP protocol in adjusting the intensity of the stream to the level of network congestion.

First AQM mechanism was proposed in 1993 by Floyd and Jacobson. Its performance is based on a drop function giving probability that a packet is

rejected. It was named Random Early Detection (sometimes Random Early Discard). The probability of packet rejection depends on the moving (weighted) average of packets queue length determined at the moment of a new packet arrival. The moving average is determined according to the actual queue length and the previous (computed upon arrival of the previous packet) moving average. This approach creates a relationship between the probability of packet loss and the queue length in the previous time moments.

The RED mechanism is very useful in maintaining a reasonable level of average queue length and queue delay. However, the selection of the parameters is not a trivial issue [24, 25]. If the parameters are not selected correctly, the performance of RED mechanism may deteriorate and then the TCP/RED system may become unstable [26]. The overview of the numerous studies on enhancing the performance of the primary RED mechanism can be found e.g. in work [15]. The analysis of the weighted average queue length implementation described in our paper [6] showed that the more detailed study of previous queue occupancy can improve the queue performance.

The AQM mechanism may be treated as a part of TCP/IP traffic control closed-loop. The feedback control system includes: (1) the TCP congestion window, which increases or decreases TCP stream as a function of losses, (2) packet buffer in a congested router, which reacts to the input stream changes, and (3) AQM mechanism, which defines the probability of the losses, which in turn, with certain delay, changes the size of congestion window.

The traditional mechanism used in automatics in feedback control system is a PI controller (Proportional-Integral controller). The extension of a PI controller is a PID controller (Proportional-Integral-Derivative controller). The PID controller is the mostly used controller in automatics, because of its simplicity, speed and availability of many efficient and simple methods of its adjusting, which in general requires the knowledge of the mathematical model of the object.

Based on the easy implementation of PI AQM controllers in real networks, a number of PI controllers have been proposed [18, 26, 27].

Theoretical studies have shown that in many cases the algorithms that use the fractional order differential-integral calculus work much better than their equivalents, based on the classical differential-integral calculus. Controllers of variable, fractional order ( $PI^\alpha D^\beta$ ) are the generalization of the classical PID controllers. The first application of the fractional order PI controller as a AQM policy in fluid flow model of a TCP connection was presented in [17].

Our papers [7–9] describe how to use the response from  $PI^\alpha$ (non-integer integral order) $D^\beta$ (non-integer derivative order) to calculate the probability of packet loss. This probability is described by a formula:

$$p_i = \max\{0, -(K_P e_k + K_I \Delta^\alpha e_k + K_D \Delta^\beta e_k)\} \quad (1)$$

where  $K_P, K_I, K_D$  are tuning parameters,  $e_k$  is the error in current moment  $k$ : the difference between desired and actual queue length, and  $\alpha$  and  $\beta$  are non integer integral and derivative orders.

The Fractional Order Derivatives and Integrals (FOD/FOI) definitions unify the definition of derivative and integral to one differintegral definition. The most

popular formulas to calculate differintegral numerically are Grunwald-Letnikov (GrLET) formula and Riemann-Liouville formulas (RL) [5, 19, 21].

Differintegral is a combined differentiation/integration operator. The  $q$ -differintegral of  $f$ , denoted by

$$\Delta^q f \quad (2)$$

is the fractional derivative (for  $q > 0$ ) or fractional integral (if  $q < 0$ ). If  $q = 0$ , then the  $q$ -th differintegral of a function is the function itself.

If order of differintegral is greater than 0 then we calculate the derivative. If the order is smaller than 0 then we calculate integral.

In the case of discrete systems (in the active queue management, packet drop probabilities are determined at discrete moments of packet arrivals) there is only one definition of differ-integrals of non-integer order. This definition is a generalization of the traditional definition of the difference of integer order to the noninteger order and it is analogous to a generalization used in Grunwald-Letnikov (GrLET) formula.

For a given sequence  $f_0, f_1, \dots, f_j, \dots, f_k$

$$\Delta^q f_k = \sum_{j=0}^k (-1)^j \binom{q}{j} f_{k-j} \quad (3)$$

where  $q \in R$  is generally a non-integer fractional order,  $f_k$  is a differentiated discrete function and  $\binom{q}{j}$  is generalized Newton symbol defined as follows:

$$\binom{q}{j} = \begin{cases} 1 & \text{for } j = 0 \\ \frac{q(q-1)(q-2)\dots(q-j+1)}{j!} & \text{for } j = 1, 2, \dots \end{cases} \quad (4)$$

Articles [7–9] show that using the non-integer order  $PI^\alpha D^\beta$  controller as AQM mechanism has advantages over the standard RED mechanism and improves the router performance. However, the disadvantage of this solution is the increase in computational complexity of the algorithm to determine packet loss probability. The standard RED algorithm has a constant computational complexity. This complexity does not depend on the queue behavior and mechanism parameters. The non-integer order  $PI^\alpha D^\beta$  algorithm has a computational complexity of the order  $k$  and this creates an increase in response time of the controller.

Experiments described in [7–9] have shown that the calculation time of the dropping packet probability is several times longer for PID controller compared to RED.

RED is one of the simplest AQM mechanism. The modifications of RED mechanism are often more complex. Some of them allow the adaptive setting of AQM parameters (e.g. ARED [13]). The adaptive mechanisms require the complicated calculations. Some mechanisms have involved the reduction in the bandwidth of aggressive streams. These mechanisms require a multiple packet draws from the queue (CHOKe algorithm family [4, 11, 22]) and sometimes the

construction of complex structures for storing auxiliary data [12]. The problem of computational complexity of these algorithms is significant especially for broadband computer networks. A delayed decision about a possible packet loss can cause a significant reduction of the transmission speed. There are articles making an attempt to improve the efficiency of AQM mechanisms (i.e. [2, 20]). However, they only describe the performance of the network transmission completely ignoring the problem of delays caused by the AQM mechanisms in themselves.

The goal of this article was to reduce the dropping packed decisions time by (i) the decrease of the number  $k$  of elements in the sum in Eq. (3) by removing from the sum the oldest queue lengths, and (ii) use the parallel GPU architecture to shorten the calculation time.

To the best of our knowledge, there are no analyses of the GPU accelerated AQM implementation.

This article describes the influence of GPU architecture on the calculation time of the probability of packet loss in  $PI^\alpha D^\beta$  controller. We implemented our algorithm on a software-based router and compare obtained response time on three types of GPU and CPU.

The remainder of the paper is organized as follows: Sect. 2 shortly describes the CUDA technology and the advantages of GPU based calculations. The section gives also some examples of algorithms implemented on GPU. Section 3 describes the GPU accelerated  $PI^\alpha D^\beta$  controller and discusses obtained results. Some conclusions are presented in Sect. 4.

## 2 CUDA Technology

NVidia's Computed Unified Device Architecture (CUDA) is a parallel computing platform which allows to use GPU for time optimization of the code. CUDA gives a programmer a tool to use of parallelism and simple program mapping to the GPU code. CUDA is designed to solve problems that can be divided into tens, hundreds and thousands of parallel calculations.

During last few years the calculations on the GPU has been used in many scientific fields. There are many examples confirming the effectiveness of the CUDA implementation, presented e.g. in [10, 16]. Paper [28] shows the practical construction of a system using the GPU to ensure the realtime operation. GPU offers many more implementation choices. Several effective techniques for implementation of vectors and vector-matrix multiplication in CUDA were investigated e.g. using single and double precision or using GPU cache memory and calculation without cache memory [3]. The article [16] shows the programmer tools and technics based on high-level languages.

Article [14] shows that the transfer of data between the CPU and the GPU introduces delays. In this transfer we need to copy the entire block of memory from the CPU to the GPU. The high-level language Python uses a list to store the block of memory. In the case of lists, data block can be placed in different parts of memory. Hence, before sending to GPU, data must be stored in memory as an array (continuous memory block). Multiple lists into arrays conversions may cause additional delays [23].

Our paper [10] described the GPU implementation of the Fluid-Flow approximation method. This implementation used the Python language. Access to nVidia's CUDA parallel computation API [1] was obtained using PyCUDA. The NumPy (library) model allows us to use the arrays. Our results presented in Sect. 3 are obtained with the use of the mentioned above tools. The Listing 1 presents the integral (derivative) calculation in CUDA.

```
# CUDA calculating module.
# Computation the sum of elements in the array
# CUDA calculating module.
# Multiplication the coefficients and queue occupancy history
import pycuda.autoinit
import pycuda.driver as drv
from pycuda.compiler import SourceModule
import pycuda.gpuarray as gpuarray
import numpy as np

#CUDA modules

# CUDA calculating module.
# Multiplication the coefficients and queue occupancy history
# dest - results array
# queueHistory
# coefficients
# n-number of multiplication(size of queueHistory or coefficients)

multiplyModule = SourceModule("""
    __global__ void multiplyKernel
    (float *dest, float * queueHistory,
    float * coefficients, int * n)
    {
        //calculate index of element in array
        const int i = blockDim.x * blockIdx.x + threadIdx.x;
        if(i < n[0])
        {
            //multiply element in queue history by coefficient
            dest[i] = queueHistory[i] * coefficients[i];
        }
    }
    """)

multiplyKernel = multiplyModule.get_function("multiplyKernel")

# CUDA calculating module.
# Computation the sum of elements in the array
```



```

# dest - results array
# vectorToSum - array with data to sum
# n - numer of additions (size of vectorToSum)

additionModule = SourceModule("""
    __global__ void additionKernel
    (float *dest, float * arrayToSum, int * n)
    {
        //calculate index of element in array
        const int i = blockDim.x * blockIdx.x + threadIdx.x;
        if(i < n[0])
        {
//CUDA array addition core function - sum of array elements
            atomicAdd(dest, arrayToSum[i]);
        }
    }
    """)
additionKernel = additionModule.get_function("additionKernel")

coefficientsVector = np.array()
queueOccupancyHistory = np.array()

blocksize = len(queueOccupancyHistory)
gridsize = 1

# create result array for CUDA multiplications

occupancyHistoryMultiplyByCoefficientVector
    = np.zeros(len(queueOccupancyHistory ))

# calculate number of CUDA block and CUDA grid size

if blocksize > multiplyKernel.max_threads_per_block:
    blocksize = multiplyKernel.max_threads_per_block
    gridsize = int(math.ceil(len(queueOccupancyHistory )
        /float(blocksize)))

# store queue occupancy history length in numpy array

queueOccupancyHistoryLength
    = np.array([len(queueOccupancyHistory)])

# send data to CUDA and run multiplications on CUDA
# data:
# occupancyHistoryMultiplyByCoefficientVector - results array
# queueOccupancyHistory

```

```

# coefficientsVector
# queueOccupancyHistoryLength - number of computations

multiplyKernel(
    drv.Out(occupancyHistoryMultiplyByCoefficientVector),
    drv.In(queueOccupancyHistory),
    coefficientsVector, drv.In(queueOccupancyHistoryLength ),
    block=(blocksize,1,1), grid=(gridsize,1))

# create result array for CUDA additions
result = np.array([0])

# send data to CUDA and run computations (additions in CUDA)
# data:
# result - results array
# occupancyHistoryMultiplyByCoefficientVector
# - data get from previous CUDA computations
# queueOccupancyHistoryLength - number of computations

additionKernel(
    drv.Out(result),
    drv.In(occupancyHistoryMultiplyByCoefficientVector),
    drv.In(queueOccupancyHistoryLength ),
    block=(blocksize,1,1), grid=(gridsize,1))

#variable result contains calculated I or D term

```

**Listing 1.** Implementation of the coefficient multiplication and adding vector of results in CUDA

### 3 The Packet Dropping Probability Function (Based on $PI^\alpha D^\beta$ Controller) Accelerated on GPU

A simple iterative implementation of  $PI^\alpha D^\beta$  controller results in increase of delays in controller response. These delays may result in the lower performance of network transmissions. Our main goal is the acceleration of the controller response. The reduction of calculation time is possible in two ways: reducing the number of PID calculations and using the parallel GPU architecture.

Our work consisted of the implementation of the non-integer order  $PI^\alpha D^\beta$  controller and testing the algorithm in the real software router. In experiments we changed the architecture of the computers. In the research we used 3 different GPU architectures:

1. NVIDIA GeForce GT 540M
2. NVIDIA GeForce GTX 1060
3. NVIDIA Quadro K2000

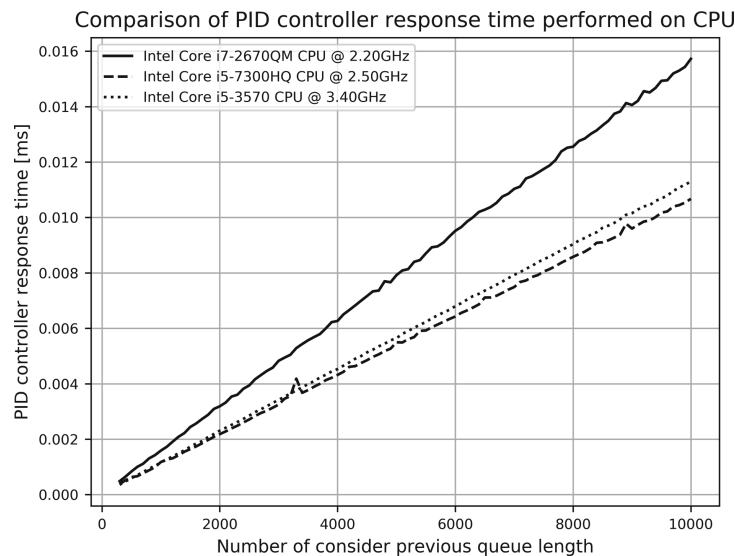
We also consider 3 different computer configurations:

1. Intel Core i5-3570 CPU @ 3.40 GHz
2. Intel Core i7-2670QM CPU @ 2.20 GHz
3. Intel Core i5-7300HQ CPU @ 2.50 GHz

We also changed the calculations by removing the oldest events from the history of the queue, i.e. limiting the horizon of differentiation/integration operator.

We focus on the relationship between the number of events in historical queue occupancy (value  $k$ , see Eq. 3), computer hardware (especially the GPU) and response time of controller. The response time of the non-integer order  $PI^\alpha D^\beta$  controller is a necessary time to complete the controller calculations (see Eq. 1).

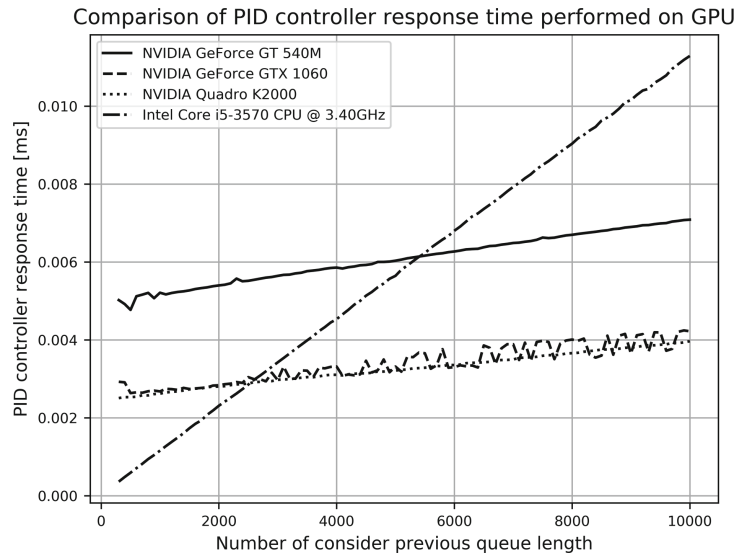
Figure 1 presents the  $PI^\alpha D^\beta$  controller response time according to the number of events in queue occupancy history considered in the calculations. Obtained results meet our expectations. The worst results were obtained in the case of the computer with an old generation i7 processor (4 cores, 8 threads, base clock 2.3, boost clock 3.10 GHz). The better results were obtained for computers with i5 processors. These results were similar because of the same processor specifications (4 cores, boost clocks between 3.5 and 3.8 GHz).



**Fig. 1.** Comparison of  $PI^\alpha D^\beta$  controller response time performed on CPU

In all of these cases increase in the number of terms of the sum in  $PI^\alpha D^\beta$  probability calculations brings to increase in response time of the  $PI^\alpha D^\beta$  controller. For a large number of terms, long response time may affect the incorrect (to slow) action of the controller. This problem can be solved by using GPU.

Figure 2 presents the same relationship as Fig. 1 but calculations (multiplication and addition) were performed on the GPU (as a comparison we add one CPU result). These results show that the response time of the  $PI^\alpha D^\beta$  controller does



**Fig. 2.** Comparison of  $PI^\alpha D^\beta$  controller response time performed on GPU

not increase as fast as on CPU. Obtained results show that the GPU implementation of the controller demonstrates the advantage when the size of the queue occupancy history exceeds (depends on GPU model) about 2000 items.

GPU introduces an additional delay associated with the transfer data between the CPU and the GPU. Figure 3 presents the relationship between the transfer time and the number of sent data. The transfer time is small and does not strongly depend on computer architecture and type of GPU. The time of transfer between GPU and CPU is about 12% of the total calculation time and does not depend significantly on the size of the transmitted data. Table 1 presents more detailed results.

**Table 1.** The times of the transfer between CPU and GPU

| Nb of terms | Times [ms]                   |              | [%]   |
|-------------|------------------------------|--------------|-------|
|             | Transfer between CPU and GPU | PID response |       |
| 1000        | 3.05400E-4                   | 2.47440E-3   | 12.34 |
| 2000        | 3.05399E-4                   | 2.52970E-3   | 12.07 |
| 3000        | 3.10400E-4                   | 2.52109E-3   | 12.31 |
| 4000        | 3.21099E-4                   | 2.55230E-3   | 12.58 |
| 5000        | 2.98499E-4                   | 2.53129E-3   | 11.79 |
| 6000        | 2.81999E-4                   | 2.54939E-3   | 11.06 |
| 7000        | 3.17300E-4                   | 2.57530E-3   | 12.32 |
| 8000        | 3.07000E-4                   | 2.58740E-3   | 11.86 |
| 9000        | 3.10399E-4                   | 2.60139E-3   | 11.93 |
| 10000       | 3.22699E-4                   | 2.59440E-3   | 12.43 |

Due to this technological delay it is not favourable to use GPU for short sequences. For long sequences, the advantages of GPU are more visible.

A data which are transferred to the GPU must have a form of an array (see Sect. 2). Regular sets of data are usually stored (in Python) in the form of lists and tuples. Therefore, each list has to be converted before transferring to the GPU. This conversion unnecessarily increases the response time of the controller. Moreover, the increase in the size of converted data results in conversion's time increase (see Table 2). Fortunately, additional Python modules (i.e. NumPy) allow us to use arrays. This modified approach (using arrays instead of lists) is a bit more complex, but it allows to avoid the unnecessary conversions. The complexity of this approach is a consequence of difficulties in selection of elements to be transferred. The results described below demonstrate that in spite of its complexity the proposed approach gives better results than the method using lists into arrays conversions.

**Table 2.** The times of the conversion from list to array

| Nb of terms | Times [ms]               |              | [%]      |
|-------------|--------------------------|--------------|----------|
|             | Conversion list to array | PID response |          |
| 1000        | 8.59000E-05              | 2.47440E-03  | 3.47155  |
| 2000        | 1.67200E-04              | 2.52970E-03  | 6.60948  |
| 3000        | 2.50000E-04              | 2.52109E-03  | 9.91635  |
| 4000        | 3.32800E-04              | 2.55230E-03  | 13.03922 |
| 5000        | 4.15700E-04              | 2.53129E-03  | 16.42246 |
| 6000        | 4.96900E-04              | 2.54939E-03  | 19.49094 |
| 7000        | 5.81200E-04              | 2.57530E-03  | 22.56824 |
| 8000        | 6.64100E-04              | 2.58740E-03  | 25.66669 |
| 9000        | 7.47000E-04              | 2.60139E-03  | 28.71542 |
| 10000       | 8.29700E-04              | 2.59440E-03  | 31.98042 |

Figures 4 and 5 present the influence of proposed approach on the response time of the  $PI^\alpha D^\beta$  controller (GPU and CPU implementations). These results confirm that our algorithm reduces the response time of the  $PI^\alpha D^\beta$  controller for calculations performed on the GPU. It was to be expected that this approach increases the time for calculations performed on the CPU.

Figure 6 presents the comparison of the response times of the  $PI^\alpha D^\beta$  controller obtained for two data format (float32 and float64). It was to be expected that the data format does not significantly influence the controller response time.

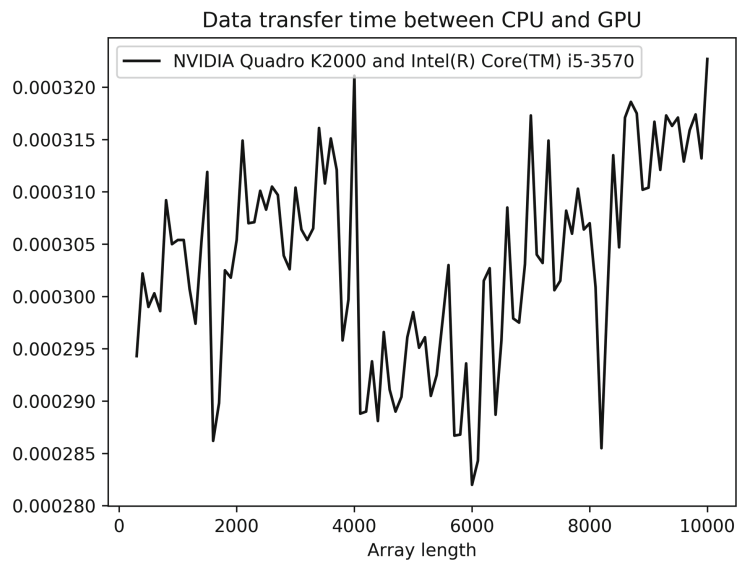


Fig. 3. Transfer between CPU and GPU times [ms]

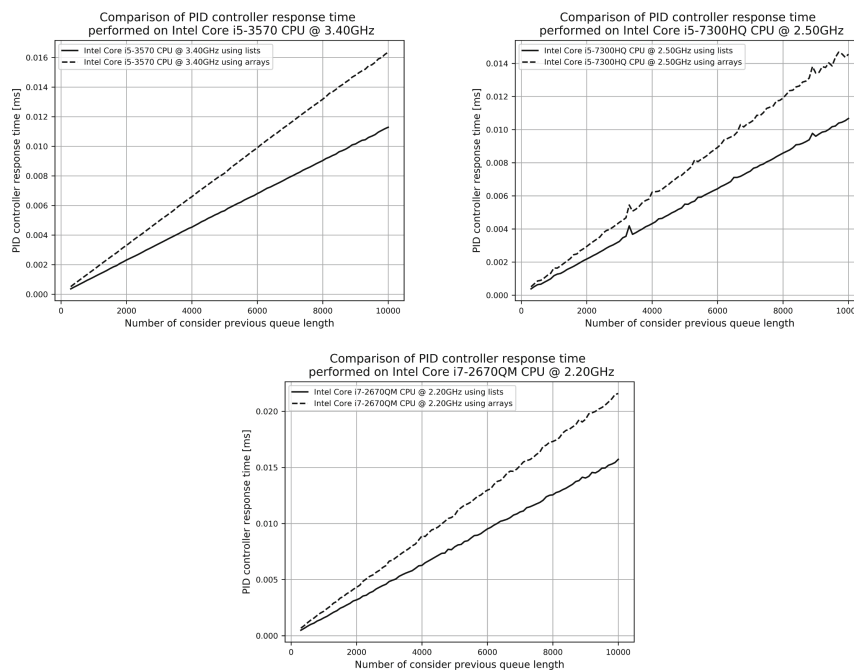
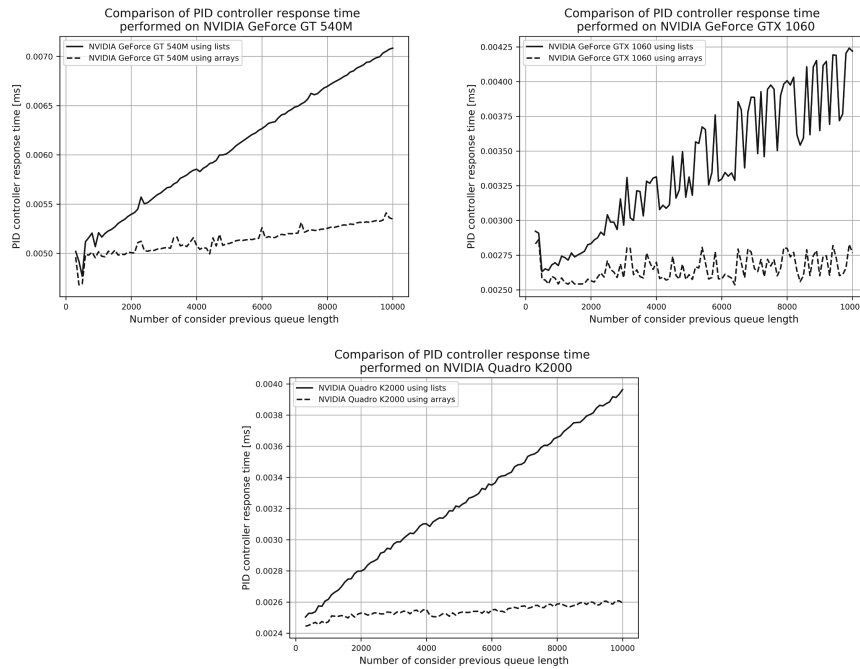
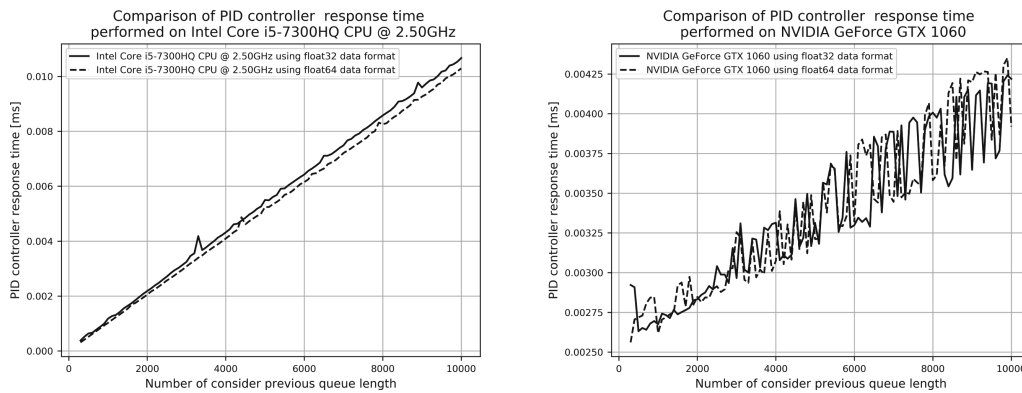


Fig. 4. Comparison of  $PI^\alpha D^\beta$  controller response time performed on CPUs with implemented arrays and lists



**Fig. 5.** Comparison of  $PI^\alpha D^\beta$  controller response time performed on GPUs with implemented arrays and lists



**Fig. 6.** Comparison of  $PI^\alpha D^\beta$  controller response time performed on GPUs and CPUs with float32 and float64 data format

## 4 Summary

The Internet Engineering Task Force (IETF) organization recommends that IP routers should use the active queue management mechanism (AQM). The basic algorithm for AQM is the RED algorithm. There are many modifications and improvements of the RED mechanism. One of these improvements is the calculation of the probability of packet loss using a  $PI^\alpha D^\beta$  controller. Our previous

work has shown the advantage of this solution. Unfortunately, this process is time-consuming.

Our paper introduces a new way of non-integer order  $PI^\alpha D^\beta$  controller calculations - using CUDA technology.

In our experiments we calculate the response time of the controller. This response time is used to calculate the packet loss probability. The results obtained using the GPU architecture were compared to the time of calculation obtained using the CPU architecture. We used three different CPU and three GPU architectures. Our results confirm that using GPU shorten the response time of a  $PI^\alpha D^\beta$  controller.

The second aspect discussed in our paper is the number of the sum terms taken into account during non-integer integral or non-integer derivative calculations. In the case of decrease in the number of terms, we don't take into account the oldest queue lengths. These research confirms that the use of GPU is efficient for long sequences of historical queue lengths. In our future work we will try to evaluate the influence of the number of terms  $k$  on  $PI^\alpha D^\beta$  controller. We expect that it depends on the nature of incoming traffic, namely on the degree of its self-similarity and on its long term autocorrelation.

We believe that the presented here results confirm the benefits of the utilisation of the  $PI^\alpha D^\beta$  controller.

## References

1. CUDA programming guide (2012)
2. Al-Faiz, M.Z., Sabry, S.S.: Optimal linear quadratic controller based on genetic algorithm for TCP/AQM router. In: International Conference on Future Communication Networks (2012)
3. Bell, N., Garland, M.: Implementing sparse matrix-vector multiplication on throughput-oriented processors. In: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis (2009)
4. Chhabra, P., Chuig, S., Goel, A., John, A., Kumar, A., Saran, H., Shorey, R.: Xchoke: malicious source control for congestion avoidance at internet gateways. In: 10th IEEE International Conference on Network Protocols (2002)
5. Leszczyński, J., Ciesielski, M.: A numerical method for solution of ordinary differential equations of fractional order. In: Wyrzykowski, R., Dongarra, J., Paprzycki, M., Waśniewski, J. (eds.) PPAM 2001. LNCS, vol. 2328, pp. 695–702. Springer, Heidelberg (2002). [https://doi.org/10.1007/3-540-48086-2\\_77](https://doi.org/10.1007/3-540-48086-2_77)
6. Domańska, J., Domański, A., Augustyn, D., J. Klamka: A RED modified weighted moving average for soft real-time application. Int. J. Appl. Math. Comput. Sci. **24**(3), 697–707 (2014)
7. Domańska, J., Domański, A., Czachórski, T., Klamka, J.: Use of a non integer order PI controller to active queue management mechanism. Int. J. Appl. Math. Comput. Sci. **26** (2016)
8. Domański, A., Domańska, J., Czachórski, T., Klamka, J.: Self-similarity traffic and AQM mechanism based on non-integer order  $PI^\alpha D^\beta$  controller. In: Gaj, P., Kwiecień, A., Sawicki, M. (eds.) CN 2017. CCIS, vol. 718, pp. 336–350. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-59767-6\\_27](https://doi.org/10.1007/978-3-319-59767-6_27)



9. Domański, A., Domańska, J., Czachórski, T., Klamka, J., Szyguła, J.: The AQM dropping packet probability function based on non-integer order  $PI^\alpha D^\beta$  controller. In: Ostalczyk, P., Sankowski, D., Nowakowski, J. (eds.) RRNR 2017. LNEE, vol. 496, pp. 36–48. Springer, Cham (2019). [https://doi.org/10.1007/978-3-319-78458-8\\_4](https://doi.org/10.1007/978-3-319-78458-8_4)
10. Domański, A., Domańska, J., Czachórski, T.: GPU-accelerated fluid flow approximation of the Active Queues Management algorithms. *Theor. Appl. Inf.* (2013)
11. Eshete, A., Jiang, Y.: Generalizing the choke flow protection. *Comput. Netw.* **57**(1), 147–161 (2013)
12. Feng, W., Shin, K., Kandlur, D., Saha, D.: The BLUE active queue management algorithms. *IEEE/ACM Trans. Netw.* (2002)
13. Floyd, S., Gummadi, R., Shenker, S.: Adaptive RED: an algorithm for increasing the robustness of RED's active queue management. *IEEE Trans. Control Syst. Technol.* (2001)
14. Fujii, Y., Azumi, T., Nishio, N., Kato, S., Edahiro, M.: Data transfer matters for GPU computing. In: *Parallel and Distributed Systems (ICPADS)* (2013)
15. Hassan, M., Jain, R.: *High Performance TCP/IP Networking*. Pearson Education Inc. (2004)
16. Klöckner, A., Pinto, N., Lee, Y., Catanzaro, B., Ivanov, P., Fasih, A.: GPU runtime code generation for high-performance computing. *Parallel Comput.* (2009)
17. Krajewski, W., Viaro, U.: On robust fractional order PI controller for TCP packet flow. In: *BOS Conference: Systems and Operational Research*, Warsaw, Poland, September 2014
18. Michiels, W., Melchor-Aguilar, D., Niculescu, S.: Stability analysis of some classes of TCP/AQM networks. *Int. J. Control* **79**, 1136–1144 (2006)
19. Miller, K., Ross, B.: *An introduction to the fractional calculus and fractional differential equations*. Wiley, New York (1993)
20. Pibiri, G., Goldrick, C.M., Huggard, M.: Enhancing AQM performance on wireless networks. In: *IFIP Wireless Days* (2012)
21. Podlubny, I.: *Fractional differential equations*. Academic Press, San Diego (1999)
22. Pan, R., Balaji Prabhakar, K.P.: Estatles AQM scheme for approximating fair bandwidth allocation. *IEEE INFOCOM* (2000)
23. Sanders, J., Kandrot, E.: *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Addison-Wesley Professional (2010). ISBN 0131387685
24. Sawicki, M., Kwiecień, A.: Unexpected anomalies of isochronous communication over USB 3.1 gen 1. *Comput. Stand. Interfaces* (2017)
25. Tan, L., Zhang, W., Peng, G., Chen, G.: Stability of TCP/RED systems in AQM routers. *IEEE Trans. Autom. Control* **51**(8), 1393–1398 (2006)
26. Unal, H., Melchor-Aguilar, D., Ustebay, D., Niculescu, S.I., Ozbay, H.: Comparison of PI controllers designed for the delay model of TCP/AQM. *Comput. Commun.* **36**, 1225–1234 (2013)
27. Ustebay, D., Ozbay, H.: Switching resilient pi controllers for active queue management of TCP flows. In: *Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control*, pp. 574–578 (2007)
28. Wodziński, M., Krzyżanowska, A.: Sequential Classification of Palm Gestures Based on Algorithm and MLP Neural Network for Quadrocopter Control. *Metrol. Meas. Syst.* (2017)

## **Rozdział 7**

### **Combined diffusion approximation - simulation model of AQMs transient behaviour**



# Combined diffusion approximation–simulation model of AQM’s transient behavior<sup>☆</sup>

Dariusz Marek<sup>a</sup>, Adam Domański<sup>a</sup>, Joanna Domańska<sup>b</sup>, Tadeusz Czachórski<sup>b</sup>, Jerzy Klamka<sup>b</sup>, Jakub Szygula<sup>a,\*</sup>

<sup>a</sup> Faculty of Automatic Control, Electronics and Computer Science, Department of Distributed Systems and Informatic Devices, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland

<sup>b</sup> Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, ul. Bałtycka 5, 44-100 Gliwice, Poland

## ARTICLE INFO

### Keywords:

Diffusion approximation  
AQM  
Congestion control  
Dropping packets  
Fractional Calculus  
Non-integer order  $PI^n$  controller  
G/G/1/N queueing model

## ABSTRACT

The article introduces an approach combining diffusion approximation and simulation ones. Furthermore, it describes how it can be used to evaluate active queue management (AQM) mechanisms. Based on the obtained queue distributions, the simulation part of the model decides on package losses and modifies the flow intensity sent by the transmitter. The diffusion is used to estimate queue distributions and the goal of the simulation part of the model is to represent the AQM mechanism. On the one hand, the use of the diffusion part considerably accelerates the performance of the whole model. On the other hand, the simulation increases the accuracy of the diffusion part. We apply the model to compare the performance of fractional order  $PI^n$  controller used in AQM with the performance of RED, a well known active queue management mechanism.

## 1. Introduction

The efficiency of the TCP protocol depends mainly on the queue management algorithm used in a router. There are distinguished two fundamental principles of network packet management. In passive queue management, packets coming to a buffer are rejected only if the buffer is fully occupied. The active mechanisms are based on the preventive packet dropping when there is still a place to store them in the queue. The packets are dropped randomly following the assumed probability loss function. This approach enhances the throughput and fairness of the link-sharing and eliminates global synchronization. The primary active queue management algorithm (AQM) is Random Early Detection (RED), primarily proposed in 1993 by Sally Floyd and Van Jacobson [1]. Several works studied the impact of various parameters on the RED performance [2,3], and many variations of AQM mechanism were developed to improve its performance, e.g. [4,5]. In order to evaluate the AQMs’ performance properly, it is necessary one create realistic models of them.

### 1.1. Our contribution

The aim of this article is to create a model combining diffusion approximation and simulation. Such a model enables us to take advantages of both approaches. Diffusion approximation is suitable to analyze transient states and therefore accelerates the performance of the whole

model. Including the simulation in the model will increase the accuracy of the diffusion part. The proposed model of a TCP connection having a bottleneck router with AQM policy is a combination of diffusion and simulation model. We use diffusion approximation to estimate the distribution of the queue length. Based on the distribution, the average queue length is calculated.

Depending on the computed average queue length, the simulation part of the model decides the packet rejection. In the case of active queue management, the decision of dropping packets is random and depends on queue occupancy and dropping packet probability function. The decision of packet rejection reduces by half the intensity  $\lambda$  of the traffic source. In the absence of a loss, the intensity of the source increases linearly. This mechanism may be seen as a part of the closed-loop control of TCP/IP traffic intensity. The evolution of the source intensity can be illustrated as follows:

$$\lambda = \begin{cases} \lambda + \zeta & \text{if } AQM \text{ decides no loss} \\ \frac{\lambda}{2} & \text{if } AQM \text{ decides a loss} \end{cases} \quad (1)$$

where  $\zeta$  is assumed a constant increase in the intensity of the source. Such source behavior is a simplified model of the TCP NewReno algorithm [6].

Using our combined model, we investigate the behavior of fractional order controller  $PI^n$  applied to control Internet traffic supervised by

<sup>☆</sup> This research was financed by the National Science Center project no. 2017/27/B/ST6/00145.

\* Corresponding author.

E-mail address: [jakub.szygula@polsl.pl](mailto:jakub.szygula@polsl.pl) (J. Szygula).

TCP transport protocol. We also investigate the influence of parameters of the controller on the queue length and evolution of the TCP congestion window. The performance of fractional order  $PI^\eta$  controller is compared to the performance of RED.

The remainder of the paper is organized as follows: Section 2 reminds works related to this article. Section 3 summarizes the basics of the diffusion approximation. In this section we present two used networks station models:

unlimited queue: G/G/1 and limited queue: G/G/1/N. Section 4 presents theoretical bases for  $PI^\eta$  controller, next used in proposed combined diffusion approximation–simulation model. Section 5 discusses the obtained results. Concluding remarks are presented in Section 6.

## 2. Related works

The diffusion approximation has many practical applications including the modeling of financial market [7]. It has been popular as a tool for road congestion analysis [8]. It is useful, for example, in risk assessments of sea turtle populations [9]. This method has been used in the evaluation of computer systems and networks for many years, see, e.g., the bibliography in [10], a more recent discussion of the errors introduced by the method in case of G/G/1 queue applied to a traffic server performance is presented in [11]. It may also be used for other than FIFO queueing disciplines, e.g., priority [12,13]. The alternative approach of transient analysis, which may be applied in traffic control models [14,15] is fluid flow approximation [16], simpler and concerning only mean values of queues.

The solution proposed in this paper is also based on diffusion approximation and G/G/1/N queueing model like in [17]. The increase in the number of users with permanent access to the Internet has created renewed interest in the research of multi-server queueing models. In the article [18] the limits of the GI/M/n/∞ were studied.

The initial AQM mechanism RED (Random Early Detection) was proposed by IETF and was primarily described by Sally Floyd and Van Jacobson in [1]. It is based on a drop function giving the probability that a packet is rejected. The argument of this function is a weighted moving average queue length, acting as a low-pass filter. This average depends on actual queue occupancy and a previously calculated value of the weighted moving average. The packet dropping probability is based on a linear function.

The introduction of AQM mechanisms has significantly improved the quality of network transmission. However, RED has such drawbacks as low throughput, unfair bandwidth sharing, the introduction of variable latency, deterioration of network stability [19]. For these reasons, numerous propositions of improvements appeared. One of these modifications is DSRED (double-slope RED) introduced and developed in [20,21]. This solution resolves the linear packet dropping function by the drop function composed of two lines with different slopes. The paper [22] proposes the NLRED algorithm with a quadratic dropping function. In the paper [23], authors considered the polynomial function.

The article [24] describes a Proportional–Integral controller on the low-frequency dynamics. The paper [25] described a robust controller, based on a known technique for  $H^\infty$  control of systems with time delays. The  $PI$  AQM controller proposed in [24] was designed following the small-gain theorem. Easy implementation of  $PI$  AQM controllers in real networks resulted in a number of propositions [26–29]. The first application of the fractional order  $PI^\eta$  controller as an AQM policy in a fluid flow model of a TCP connection was presented in [30].

Article [31] introduces a simple diffusion model of a RED control mechanism working in an open-loop scenario. Our paper [32] describes diffusion models of a RED and a fractional order  $PI^\eta$  controller working under TCP NewReno’s control. To the best of our knowledge, there is no paper describing a combination of diffusion-simulation approaches.

## 3. Diffusion approximation

The main principle of the diffusion approximation method [33–36], is replacing the discrete process  $N(t)$  i.e. the number of customers in the queue, by a continuous diffusion process  $X(t)$ . Similarly as in the case of  $N(t)$ , the changes of  $dX(t) = X(t+dt) - X(t)$  are normally distributed with the mean  $\beta dt$  and variance  $\alpha dt$  determined by the parameters  $\beta$ ,  $\alpha$  of the diffusion equation [37,38]:

$$\frac{\partial f(x, t; x_0)}{\partial t} = \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x} \quad (2)$$

where  $f(x, t : x_0)$  is the probability density function (pdf) of the diffusion process [38]:

$$f(x, t; x_0) dx = P[x \leq X(t) < x + dx | X(0) = x_0]. \quad (3)$$

The solution of Eq. (2) is used for the evaluation of the queue distribution in the investigated. It may be applied in case when the interarrival and service times follow general distributions  $A(x)$ ,  $B(x)$ , i.e. in modeling G/G/1 and G/G/1/N service systems. Two first moments of these distributions are considered:  $E[A] = 1/\lambda$ ,  $E[B] = 1/\mu$ ,  $Var[A] = \sigma_A^2$ ,  $Var[B] = \sigma_B^2$ .

Denote also the squared coefficients of variation:  $C_A^2 = \sigma_A^2 \lambda^2$ ,  $C_B^2 = \sigma_B^2 \mu^2$ . The choice of diffusion parameters:

$$\beta = \lambda - \mu, \quad \alpha = \sigma_A^2 \lambda^3 + \sigma_B^2 \mu^3 = C_A^2 \lambda + C_B^2 \mu \quad (4)$$

### 3.1. Unlimited queue: G/G/1 station

More formal justification of diffusion approximation is in limit theorems for G/G/1 system given by Iglehart and Whittle [39–42]: If  $\hat{N}_n$  is a series of random variables derived from  $N(t)$  [10]:

$$\hat{N}_n = \frac{N(nt) - (\lambda - \mu)nt}{(\sigma_A^2 \lambda^3 + \sigma_B^2 \mu^3) \sqrt{n}} \quad (5)$$

then the series is weakly convergent (in the sense of distribution) to  $\xi$  where  $\xi(t)$  is a standard Wiener process (i.e. diffusion process with  $\beta = 0$  i  $\alpha = 1$ ) provided that  $\rho > 1$ , that means if the system is overloaded and has no equilibrium state. In the case of  $\rho = 1$  the series  $\hat{N}_n$  is convergent to  $\xi_R$ . The  $\xi_R(t)$  process is  $\xi(t)$  process limited to half-axis  $x > 0$  [42]:

$$\xi_R(t) = \xi(t) - \inf [\xi(u), 0 \leq u \leq t]. \quad (6)$$

The process  $N(t)$  is never negative, therefore  $X(t)$  should also be limited to values  $x \geq 0$ . It is done by placing at  $x = 0$  a barrier which prevents the process from moving into negative part of  $x$  axis.

One choice is to place at  $x = 0$  a reflecting barrier [43] that limits the process to a positive  $x$ -axis and is equivalent to the condition:

$$\int_0^\infty f(x, t; x_0) dx = 1$$

and:

$$\frac{\partial}{\partial t} \int_0^\infty f(x, t; x_0) dx = \int_0^\infty \frac{\partial f(x, t; x_0)}{\partial t} dx = 0. \quad (7)$$

Replacing the integrated function with the right side of the diffusion equation, we get the boundary condition corresponding to the reflecting barrier at zero [10]:

$$\lim_{x \rightarrow 0} \left[ \frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0) \right] = 0. \quad (8)$$

The solution of Eq. (3) with boundary conditions defined by Eq. (8) gives us: [43]

$$f(x, t; x_0) = \frac{\partial}{\partial x} \left[ \Phi \left( \frac{x - x_0 - \beta t}{\alpha t} \right) - e^{-\frac{2\beta x}{\alpha}} \Phi \left( \frac{x + x_0 + \beta t}{\alpha t} \right) \right], \quad (9)$$

where:  $\Phi(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$  is the PDF of standard normal distribution.

For  $(\beta < 0)$  the system converges to a steady-state:  $\lim_{t \rightarrow \infty} f(x, t; x_0) = f(x)$  and partial differential equation (2) becomes an ordinary one [10]:

$$0 = \frac{\alpha}{2} \frac{d^2 f(x)}{dx^2} - \beta \frac{d f(x)}{dx} \tag{10}$$

with solution:

$$f(x) = -\frac{2\beta}{\alpha} e^{\frac{2\beta x}{\alpha}}.$$

This formula approximates the queue at G/G/1 system [10]:

$$p(n, t; n_0) \approx f(n, t; n_0), \tag{11}$$

and at steady-state  $p(n) \approx f(n)$ ; one can also choose e.g. [37]

$$p(0) \approx \int_0^{0.5} f(x) dx, p(n) \approx \int_{n-0.5}^{n+0.5} f(x) dx, n = 1, 2, \dots, \tag{12}$$

The reflecting barrier in point  $x = 0$  causes the immediate reflecting of the process, that means  $f(0, t; x_0) = 0$ . This version of the diffusion process is a heavy-load approximation and gives good results in case of a system with utilization factor  $\rho$  close to 1, i.e. probability of the empty system is closed to 0.

This restriction is removed by another type of limit condition at  $x = 0$ : a barrier with instantaneous elementary returns (sometimes called jumps) [38]. When the diffusion process reaches  $x = 0$ , it remains there for a random time corresponding to the idle period with no customers in the queue and the returns to  $x = 1$  (it corresponds to the arrival of one packet to the system).

The time when the process is at  $x = 0$  means that the system is in the idle state. The diffusion equation becomes [38]:

$$\begin{aligned} \frac{\partial f(x, t; x_0)}{\partial t} &= \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} \\ -\beta \frac{\partial f(x, t; x_0)}{\partial x} + \lambda p_0(t) \delta(x - 1), \\ \frac{d p_0(t)}{dt} &= \lim_{x \rightarrow 0} \left[ \frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0) \right] - \lambda p_0(t), \end{aligned} \tag{13}$$

where:  $p_0(t) = P[X(t) = 0]$ .

The term  $\lambda p_0(t) \delta(x - 1)$  gives the probability density that the process is started at point  $x = 1$  at the moment  $t$ , because of the jump from the barrier. The second equation balances  $p_0(t)$ : the term  $\lim_{x \rightarrow 0} [\frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0)]$  gives the probability flow into the barrier, while  $\lambda p_0(t)$  represents the probability flow out of the barrier [10].

The approach to obtain the function  $f(x, t; x_0)$  of the process with jumps from the barrier is to express it with the use of another pdf  $\phi(x, t; x_0)$  for the diffusion process with the absorbing barrier at  $x = 0$  [17]. This process starts at  $t = 0$  from  $x = x_0$  and ends when it reaches the barrier. Such probability density function is easier to determine [44]:

$$\phi(x, t; x_0) = \frac{e^{\frac{\beta}{\alpha}(x-x_0) - \frac{\beta^2}{2\alpha}t}}{\sqrt{2\pi\alpha t}} \left[ e^{-\frac{(x-x_0)^2}{2\alpha t}} - e^{-\frac{(x+x_0)^2}{2\alpha t}} \right] \tag{14}$$

The density function of the first passage time from  $x = x_0$  to  $x = 0$  is [17]:

$$\begin{aligned} \gamma_{x_0,0}(t) &= \lim_{x \rightarrow 0} \left[ \frac{\alpha}{2} \frac{\partial}{\partial x} \phi(x, t; x_0) - \beta \phi(x, t; x_0) \right] = \\ &= \frac{x_0}{\sqrt{2\pi\alpha t^3}} e^{-\frac{\beta t + x_0}{2\alpha t}} \end{aligned} \tag{15}$$

Assume that process with the barrier with jumps starts at  $t = 0$  at a point  $x > 0$  with density  $\psi(x)$  and when it reaches the barrier, it stays there for a time defined by a density function  $l_0(x)$ , then jumps to the point  $x = 1$  and again, after certain time, returns to  $x = 0$ , stays there, jumps to  $x = 1$ , etc. The total stream  $\gamma_0(t)$  of probability mass that enters the barrier is [17]:

$$\gamma_0(t) = p_0(0)\delta(t) + [1 - p_0(0)]\gamma_{\psi,0}(t) + \int_0^t g_1(\tau)\gamma_{1,0}(t - \tau)d\tau, \tag{16}$$

where:

$$\begin{aligned} \gamma_{\psi,0}(t) &= \int_0^\infty \gamma_{\xi,0}(t)\psi(\xi)d\xi, \\ g_1(\tau) &= \int_0^\tau \gamma_0(t)l_0(\tau - t)dt. \end{aligned} \tag{17}$$

The density function of the diffusion process with instantaneous returns is [17]:

$$f(x, t; x_0) = \phi(x, t; \psi) + \int_0^t g_1(\tau)\phi(x, t - \tau; 1)d\tau. \tag{18}$$

Denote by  $\bar{f}(s)$  the Laplace transform of a function  $f(x)$ . For the Eqs. (16) and (17) there are following Laplace transforms [17]:

$$\begin{aligned} \bar{\gamma}_0(s) &= p_0(0) + [1 - p_0(0)]\bar{\gamma}_{\psi,0}(s) + \bar{g}_1(s)\bar{\gamma}_{1,0}(s), \\ \bar{g}_1(s) &= \bar{\gamma}_0(s)\bar{l}_0(s), \end{aligned} \tag{19}$$

where:

$$\begin{aligned} \bar{\gamma}_{x_0,0}(s) &= e^{-x_0 \frac{\beta + A(s)}{\alpha}}, \quad A(s) = \sqrt{\beta^2 + 2\alpha s}, \\ \bar{\gamma}_{\psi,0}(s) &= \int_0^\infty \bar{\gamma}_{\xi,0}(s)\psi(\xi)d\xi, \end{aligned} \tag{20}$$

and then:

$$\bar{g}_1(s) = [p_0(0) + [1 - p_0(0)]\bar{\gamma}_{\psi,0}(s)] \frac{\bar{l}_0(s)}{1 - \bar{l}_0(s)\bar{\gamma}_{1,0}(s)}. \tag{21}$$

Eq. (18) in terms of the Laplace transform becomes [17]:

$$\bar{f}(x, s; x_0) = \bar{\phi}(x, s; \psi) + \bar{g}_1(s)\bar{\phi}(x, s; 1), \tag{22}$$

where:

$$\begin{aligned} \bar{\phi}(x, s; x_0) &= \frac{e^{\beta(x-x_0)/\alpha}}{A(s)} \left[ e^{-|x-x_0| \frac{A(s)}{\alpha}} - e^{-|x+x_0| \frac{A(s)}{\alpha}} \right], \\ \bar{\phi}(x, s; \psi) &= \int_0^\infty \bar{\phi}(x, s; \xi)\psi(\xi)d\xi. \end{aligned} \tag{23}$$

The inverse transforms of these functions can be obtained numerically. For this purpose, Stehfest's algorithm [45] is used.

The presented above G/G/1 model assumes constant model parameters. In the case of variable model parameters, we should divide the model time into periods, where the parameters of the model are constant. In this approach the results obtained at the end of one time period serve as the initial condition for the next period.

### 3.2. Limited queue: G/G/1/N station

In the case of G/G/1/N station, the number of packets in the node is limited to  $N$ . That means that we need in our model the second barrier placed at  $x = N$ .

So in G/G/1/N model, the diffusion process is limited by two barriers. The first barrier is placed in  $x = 0$ , and the second is placed at  $x = N$ . The behavior of the process in the first barrier we described in the previous subsection. In the second barrier, when the process reaches  $x = N$ , it waits there for a time corresponding to the period when the queue is full and incoming packets are being lost and then jumps to  $x = N - 1$ . The density function of such process  $f(x, t; x_0)$  is obtained similarly to the model with infinite queue described before.

In the first step, the method determines the density  $\phi(x, t; x_0)$  of the diffusion process with two absorbing barriers at  $x = 0$  and  $x = N$ , started at  $t = 0$  from  $x = x_0$  [44]:

$$\begin{aligned} \phi(x, t; x_0) &= \frac{1}{\sqrt{2\pi\alpha t}} \sum_{n=-\infty}^\infty \left( \exp\left[ \frac{\beta x'_n}{\alpha} - \frac{(x - x_0 - x'_n - \beta t)^2}{2\alpha t} \right] \right. \\ &\quad \left. - \exp\left[ \frac{\beta x''_n}{\alpha} - \frac{(x - x_0 - x''_n - \beta t)^2}{2\alpha t} \right] \right), \end{aligned} \tag{24}$$

where:  $x'_n = 2nN, x''_n = -2x_0 - x'_n$ .

For the initial condition is defined by a function  $\psi(x)$ ,  $x \in (0, N)$ ,  $\lim_{x \rightarrow 0} \psi(x) = \lim_{x \rightarrow N} \psi(x) = 0$ , then the pdf of the process has the form:

$$\phi(x, t; \psi) = \int_0^N \phi(x, t; \xi) \psi(\xi) d\xi. \tag{25}$$

The density function  $f(x, t; \psi)$  of diffusion process with instantaneous returns from both barriers is composed of function  $\varphi(x, t; \psi)_1$  representing the influence of the initial condition  $\psi$  and the set of functions  $\varphi(x, t - \tau; 1)$ ,  $\varphi(x, t - \tau; N - 1)$  started after the jump from barrier at time  $\tau < t$ , at points  $x = 1$  and  $x = N - 1$  with intensities  $g_1(\tau)$  and  $g_{N-1}(\tau)$  [17]:

$$f(x, t; \psi) = \varphi(x, t; \psi) + \int_0^t g_1(\tau) \varphi(x, t - \tau; 1) d\tau + \int_0^t g_{N-1}(\tau) \varphi(x, t - \tau; N - 1) d\tau. \tag{26}$$

Functions  $g_1(t)$ ,  $g_N(t)$  can be expressed by means of probability densities  $\gamma_0(t)$  and  $\gamma_N(t)$  [31]:

$$g_1(\tau) = \int_0^\tau \gamma_0(t) l_0(\tau - t) dt, \tag{27}$$

$$g_{N-1}(\tau) = \int_0^\tau \gamma_N(t) l_N(\tau - t) dt,$$

where:  $l_0(x)$ ,  $l_N(x)$  are density functions of the distribution of time which the process stays at  $x = 0$  and  $x = N$ . Not that these distributions may be general and not restricted to exponential ones.

Probability densities  $\gamma_0(t)$ ,  $\gamma_N(t)$  that process enters the barrier at  $x = 0$ , or  $x = N$  at time  $t$  are [17]:

$$\gamma_0(t) = p_0(0) \delta(t) + [1 - p_0(0) - p_N(0)] \gamma_{\psi,0}(t) + \int_0^t g_1(\tau) \gamma_{1,0}(t - \tau) d\tau + \int_0^t g_{N-1}(\tau) \gamma_{N-1,0}(t - \tau) d\tau, \tag{28}$$

$$\gamma_N(t) = p_N(0) \delta(t) + [1 - p_0(0) - p_N(0)] \gamma_{\psi,N}(t) + \int_0^t g_1(\tau) \gamma_{1,N}(t - \tau) d\tau + \int_0^t g_{N-1}(\tau) \gamma_{N-1,N}(t - \tau) d\tau,$$

where:  $\gamma_{1,0}(t)$ ,  $\gamma_{1,N}(t)$ ,  $\gamma_{N-1,0}(t)$ ,  $\gamma_{N-1,N}(t)$  are the densities of first passage times of diffusion process between corresponding points indicated in the lower index, [17]:

$$\gamma_{1,0}(t) = \lim_{x \rightarrow 0} \left[ \frac{\alpha}{2} \frac{\partial \phi(x, t; 1)}{\partial x} - \beta \phi(x, t; 1) \right]. \tag{29}$$

Functions  $\gamma_{\psi,0}(t)$ ,  $\gamma_{\psi,N}(t)$  define probability densities, that the initial process, started on  $t = 0$ , at the point  $x = \xi$  with density  $\psi(\xi)$ , will end at time  $t$  by entering the barrier respectively at  $x = 0$  or  $x = N$ .

Finally, probabilities that at time  $t$  the process has value  $x = 0$  or  $x = N$  are:

$$p_0(t) = \int_0^t [\gamma_0(\tau) - g_1(\tau)] d\tau, \tag{30}$$

$$p_N(t) = \int_0^t [\gamma_N(\tau) - g_{N-1}(\tau)] d\tau.$$

Similarly to the previous model of the Section 3.1, the presented equations are transformed using Laplace transform and determine the transformed function  $\bar{f}(x, s; \psi)$ , and its original is obtained numerically. As previously, to vary with time the diffusion coefficients, we divide the time axis into subintervals with specific constant parameters, and the solution of the end of an interval gives initial conditions for the next one. The presented transient solution tends for  $t \rightarrow \infty$ , to the known steady-state solutions; therefore, models based on diffusion approximation can also refer to the steady-state. This approach was used in several models, e.g. in RED modeling [31], and its correctness was verified with simulations, e.g. [46].

#### 4. AQM mechanism based on non-integer order $PI^\eta$ controller

Recent scientific and engineering studies have shown that the dynamics of many systems can be described more precisely using

non-integer differential equations [47]. Fractional Order Derivatives and Integrals (FOD/FOI) are a natural extension of the well-known integrals and derivatives. Differintegrals of non-integer orders enable better and more precise control of dynamical processes allowing to include memory in a process. There are several definitions related to FOD/FOI. The most popular continuous definitions include: Riemann–Liouville [48], Atangana–Baleanu [49], Caputo–Fabrizio [50] and Liouville–Caputo [47]. There are also discrete solutions of Caputo definition [51] and Riemann–Liouville definitions [51,52]. Our model takes into account discrete moments of packet arrivals so we decide to use Grünwald–Letnikov discrete fractional operator [53,54].

A Proportional–Integral controller (PI controller) is a traditional mechanism used in feedback control systems. Earlier works [55,56] indicates that the non-integer order controllers have better behavior than classic ones. Authors of [57] confirm that fractional order controllers achieve better results than the best integer order controllers. There are many studies confirming the advantage of FOD over classical derivatives. The article [58] describes the use of fractional order PID controller to control the temperature in an isothermal room used in a pharmaceutical factory. The simulation results indicate that the proposed controller provides better quality of control than a classical PID controller with autotuning. The authors of the [59] also confirm that the non-integer PID controller provides better and more stable system operation than the traditional one.

The articles [60–63], describe how to use the response of the  $PI^\eta$  (non-integer integral order) to calculate the response of the AQM mechanism. The probability of packet loss is given by the formula:

$$p_l = \max\{0, -(K_p e_k + K_I \Delta^\eta e_k)\} \tag{31}$$

where  $K_p, K_I$  are tuning parameters,  $e_k$  is the error in current slot  $e_k = Q_k - Q$ , i.e. the difference between current queue  $Q_k$  and desired queue  $Q$ .

Thus, the dropping probability depends on three parameters: the coefficients for the proportional and integral terms  $K_p, K_I$  and integral order  $\eta$ .

In the active queue management, packet drop probabilities are determined at discrete moments of packet arrivals; hence the queue model should be considered as a case of discrete systems. We use Grünwald–Letnikov (GrLET) definition of the discrete differ-integrals of non-integer order. This definition is a generalization of the traditional definition of the difference of integer order to the non-integer order, and it is analogous to a generalization used in GrLET formula.

For a given sequence  $f_0, f_1, \dots, f_j, \dots, f_k$  [64,65]:

$$\Delta^\eta f_k = \sum_{j=0}^k (-1)^j \binom{\eta}{j} f_{k-j} \tag{32}$$

where  $\eta \in R$  is generally a non-integer fractional order,  $f_k$  is a differentiated discrete function, and  $\binom{\eta}{j}$  is generalized Newton symbol defined as follows:

$$\binom{\eta}{j} = \begin{cases} 1 & \text{for } j = 0 \\ \frac{\eta(\eta - 1)(\eta - 2) \dots (\eta - j + 1)}{j!} & \text{for } j = 1, 2, \dots \end{cases} \tag{33}$$

For  $\eta = 1$  we get the formula for the difference of the first order (only two coefficients are non-zero).

$$\Delta^1 x_k = 1x_k - 1x_{k-1} + 0x_{k-2} + 0x_{k-3} \dots \tag{34}$$

For  $\eta = -1$  we get the sum of all samples (the discrete integral of first order equivalent).

$$\Delta^{-1} x_k = 1x_k + 1x_{k-1} + 1x_{k-2} + 1x_{k-3} \dots \tag{35}$$

For non-integer derivative and integral order we get the weighted sum of all samples, e.g.:

$$\Delta^{-1.2} x_k = 1x_k + 1.2x_{k-1} + 1.32x_{k-2} + 1.408x_{k-3} \dots$$

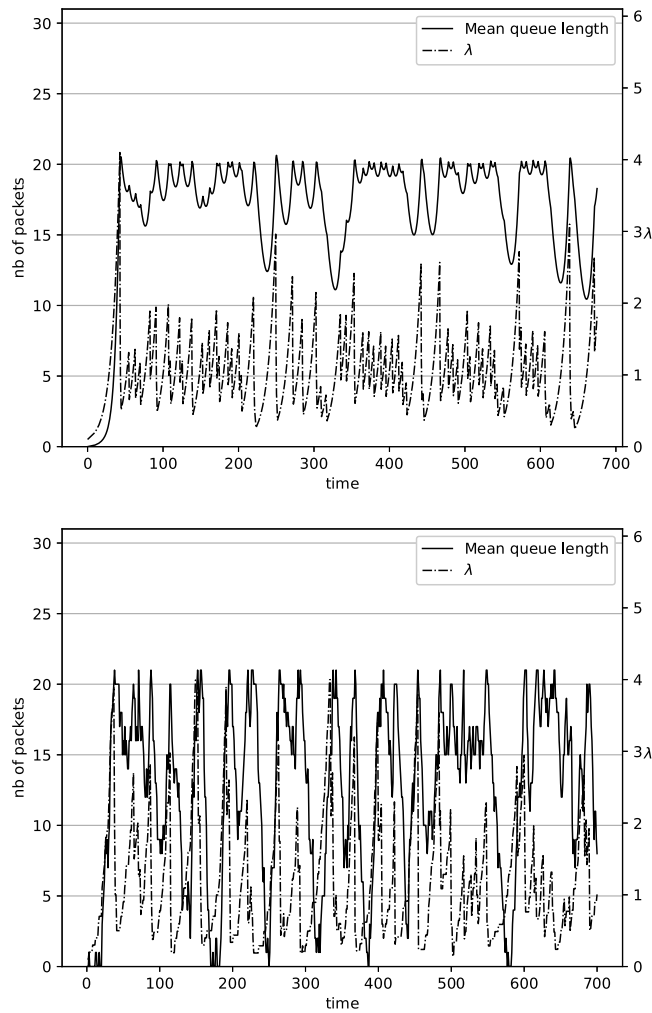
$$\Delta^{-0.8} x_k = 1x_k + 0.8x_{k-1} + 0.72x_{k-2} + 0.672x_{k-3} \dots \tag{36}$$

**Table 1**  
*PI<sup>n</sup>* controllers coefficients.

|   | $K_p$  | $K_i$  | $\eta$ | Setpoint |
|---|--------|--------|--------|----------|
| 1 | 0.0001 | 0.0004 | -1.2   | 10       |
| 2 | 0.0001 | 0.0004 | -1.0   | 10       |
| 3 | 0.0001 | 0.0014 | -0.8   | 10       |
| 4 | 0.0001 | 0.0040 | -0.4   | 10       |

**Table 2**  
*PI<sup>n</sup>* obtained average queue lengths.

| AQM                     | Diffusion | Simulation |
|-------------------------|-----------|------------|
| RED                     | 18.067    | 17.106     |
| <i>PI<sup>n</sup></i> 1 | 5.100     | 5.251      |
| <i>PI<sup>n</sup></i> 2 | 6.122     | 6.302      |
| <i>PI<sup>n</sup></i> 3 | 7.780     | 7.123      |
| <i>PI<sup>n</sup></i> 4 | 10.431    | 10.075     |

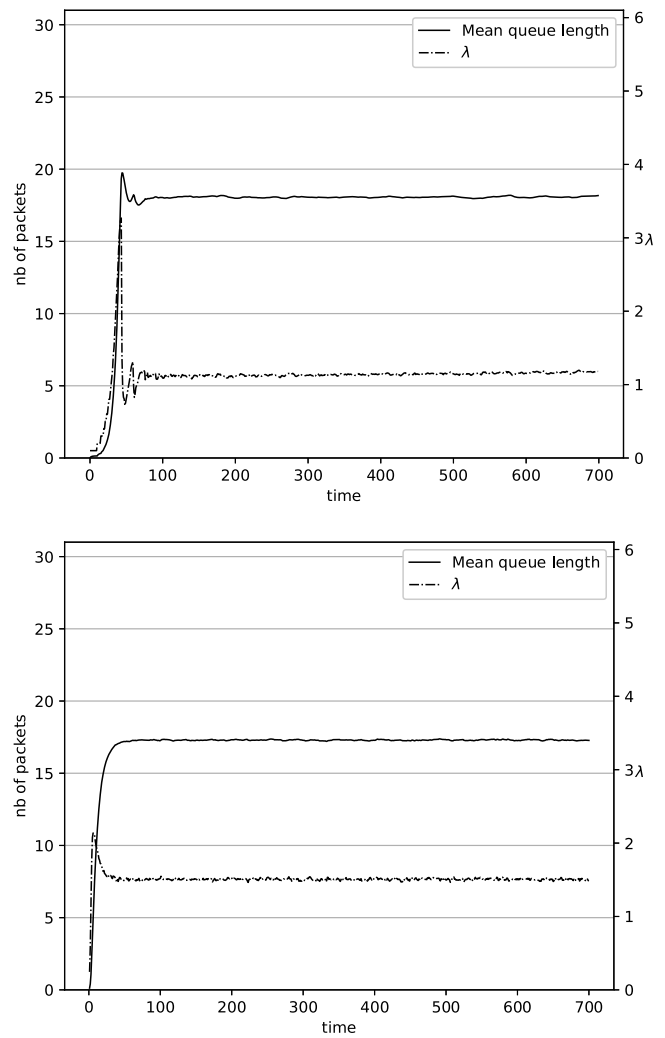


**Fig. 1.** The router mean queue length and source intensity, diffusion approximation (upper), simulation (bottom), RED algorithm.

**5. Diffusion approximation analysis of AQM performance**

The presented mixed diffusion-simulation model uses the limited queue, i.e. G/G/1/N station to obtain the time-dependent average queue length. Based on it, the simulation part of the model decides on dropping packets.

Calculations for both models (the mixed diffusion-simulation and simple simulation) were performed in Python and C. The simulations



**Fig. 2.** The router mean queue length and source intensity, diffusion approximation (upper), simulation (bottom), RED algorithm.

were done using the SimPy Python simulation packet. SimPy is released under the MIT License (free software license originating at the Massachusetts Institute of Technology). During the tests, we analyzed such parameters of the transmission with AQM as the length of the queue and changes of the source intensity  $\lambda$ .

The diffusion calculations were carried with  $\Delta t = \frac{1}{2}$  step. As a result, we obtain a queue distribution and then its mean size. On this basis, the simulation part of the model decides if the packet is rejected. The simulator of the AQM mechanism draws the value given by a uniform generator of the range  $\langle 0;1 \rangle$ , compares it with the value of the packet dropping function (which is a function of the queue length) and decides if the packet is accepted or deleted. This decision influences the new value of  $\lambda$ ; the change is made after  $\Delta t$  delay which represents the time after which the loss information is received by the TCP transmitter, and the traffic with new intensity reaches the router. The service time represents the time of a packet treatment and dispatching we assume it is exponentially distributed with constant parameter  $\mu = 1$ . The results of the mixed diffusion-simulation model are compared with pure simulation ones.

We considered two types of AQM algorithms: standard RED and algorithm based on the answer of *PI<sup>n</sup>* controller.

The queue parameters were as follows: maximal queue lengths = 30, RED parameters  $Min_{th} = 10$ ,  $Max_{th} = 20$ , *PI<sup>n</sup>* desire queue length = 10.

To obtain reliable transient state results, we repeated the experiments 10,000 times. An example of the results of a single experiment is

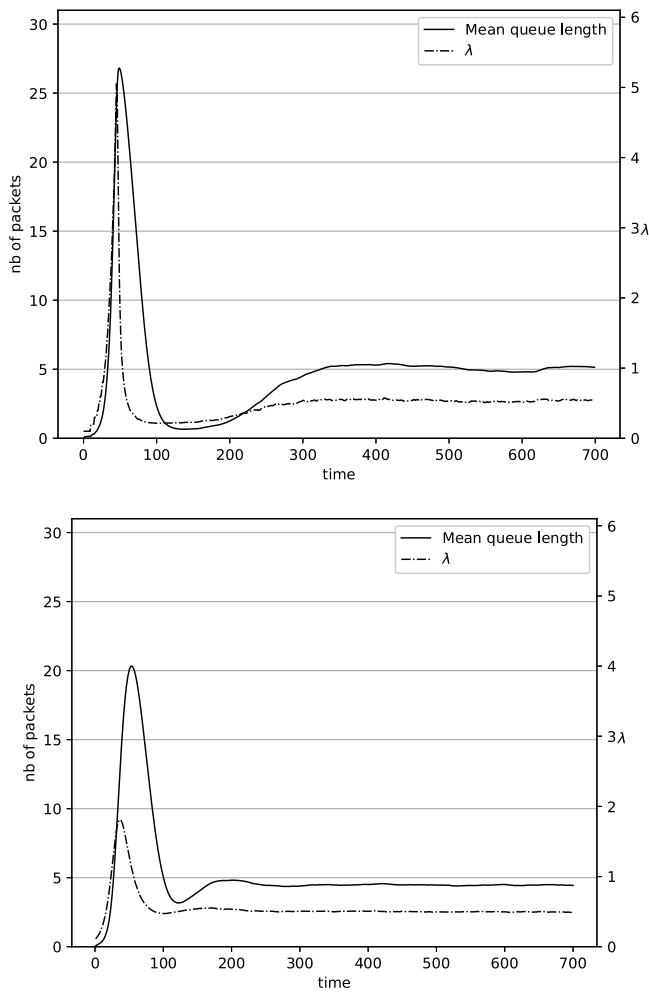


Fig. 3. The router mean queue length and source intensity, diffusion approximation (upper), simulation (bottom), 1st set of  $PI^\eta$  parameters.

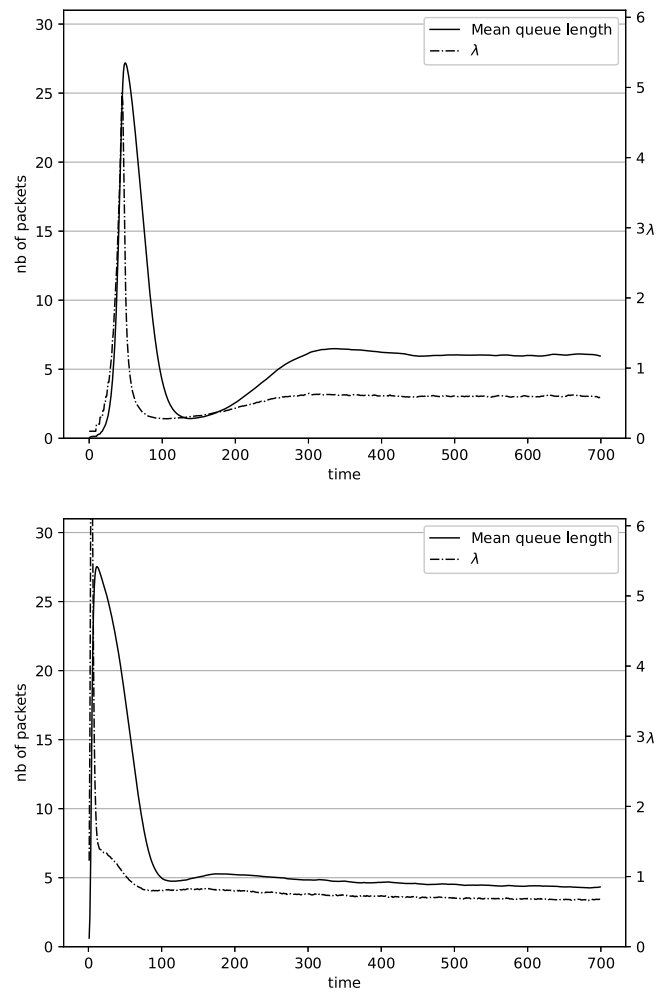


Fig. 4. The router mean queue length and source intensity, diffusion approximation (upper), simulation (bottom), 2nd set of  $PI^\eta$  parameters.

shown in Fig. 1. The calculation time for a single simulation experiment does not exceed one second. In the case of diffusion calculations it takes a few minutes. Each figure presents two plots: mean queue length with y axis on the left and source intensity with y axis on the right. The queue occupancy varies according to the change in the intensity of the source (related to the TCP mechanism).

On the first stage of the experiment, we considered the RED algorithm as AQM. Sample results of changes in the traffic intensity and queue occupancy are presented in Fig. 2. Table 2 presents the detailed results for all performed experiments.

Fig. 2 shows that the mean queue length reaches the value of 17 packets. This is in line with the results presented in Table 2. You can see in the same figure that congestion window control mechanism sets the source intensity at around 1,2.

In the next phase of the experiments, we evaluate the AQM mechanism with packet dropping probability function based on  $PI^\eta$ . We consider four different controllers. The  $PI^\eta$  controller with parameter  $\eta = -1$  denotes the classic  $PI$  controller. The Table 1 presents their parameters. The influence of the parameters on the efficiency of AQM queue was discussed in [62,66,67]. The proposed controllers differ in the effectiveness of maintaining the desired queue size.

Figs. 3, 4, 5, 6 present how the intensity of the source evaluates over the time and how it affects the queue occupancy.

The first controller (2nd row in Table 1) is the most effective (we call it the most powerful) but, it rejects a large number of packets. In

the case of this controller the mean queue length reaches the value of 5 packets (Fig. 3). The fourth type of controller (5th row in Table 1) has inverse properties. It rejects a smaller number of packets but its mean queue length reaches a bit higher value, about 10 packets. The characteristics of the second and third controller (3rd and 4th rows in Table 1) are in-between the properties of the first and the fourth one. The mean queue lengths for these controllers reach accordingly the values of 6 and 7 packets.

The detailed results confirm the outcomes presented in figures. The average queue lengths in steady-state (Fig. 2, 3, 4, 5, 6.) are identical to those presented in Table 2.

Additionally, for all AQM mechanisms (including the RED algorithm) differences in results obtained by diffusion approximation and simulations are negligible.

The influence of controller parameters on the behavior of the AQM queue is visible. The average queue length decreases (Table 2) with increasing controller fractional order (Table 1).

The advantage of diffusion approximation is the ability to consider two first moments of traffic (intensity and variation), instead of only one in case of more foreign full used fluid flow approximation. Fig. 7 presents the influence of the traffic source variation on queue behavior. The impact is relatively low. We think that it is due to the performance of TCP protocol and its mechanism of adapting the transmission speed to the network capabilities.



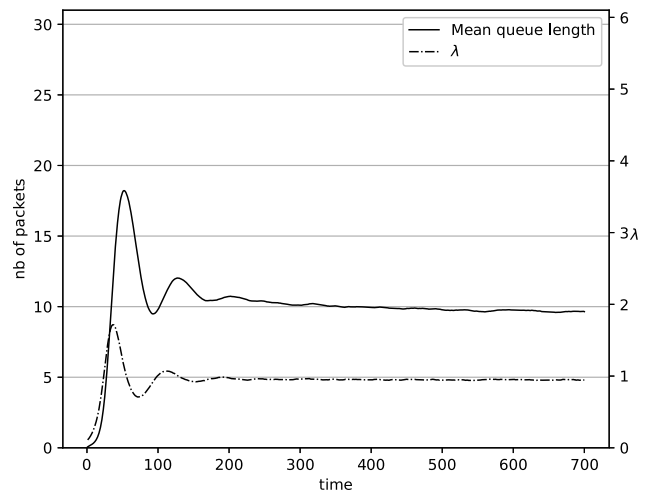
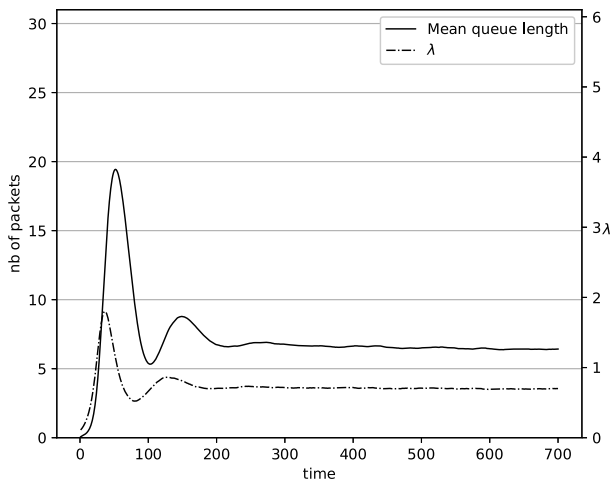
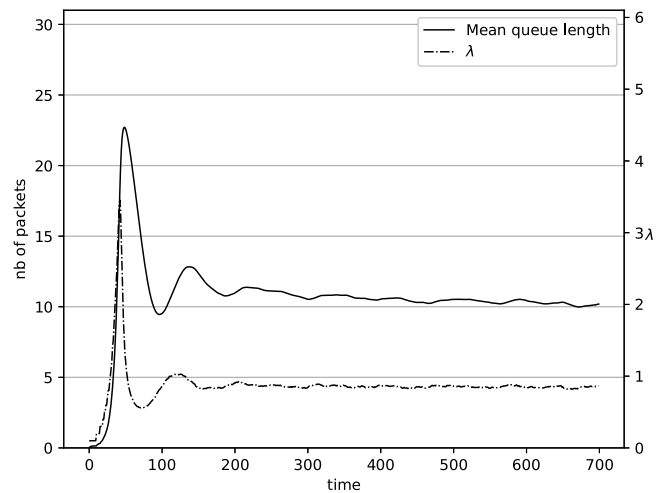
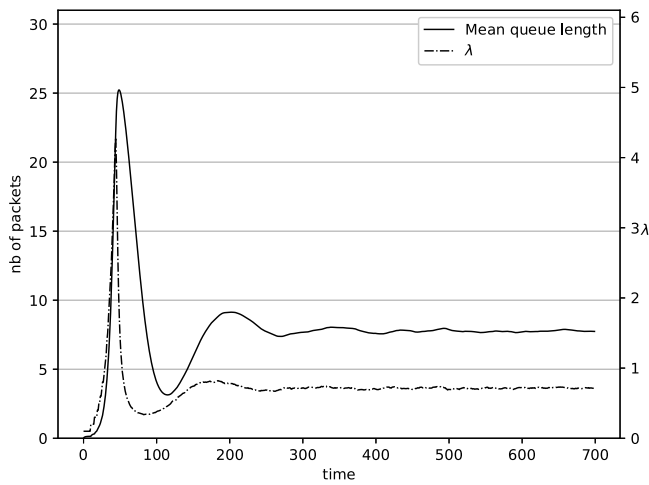


Fig. 5. The router mean queue length and source intensity, diffusion approximation (upper), simulation (bottom), 3rd set of  $PI^\alpha$  parameters.

Fig. 6. The router mean queue length and source intensity, diffusion approximation (upper), simulation (bottom), 4th set of  $PI^\alpha$  parameters.

### 6. Conclusions

This article proposes a model combining the diffusion approximation with discrete event simulation. The approximation is used to calculate the size of the queue. The advantage of this approach is a natural description of transient states for any interarrival and service time distributions. The simulation part decides packet rejection following the AQM rules. This decision affects the source intensity of  $\lambda$  in the diffusion model. We consider the RED,  $PI$  and  $PI^\alpha$  algorithms.

We compare the results obtained from the proposed mixed diffusion-simulation method with the results obtained by simulation. In both cases, the results are consistent. In the case of the  $PI^\alpha$  controller, the obtained average queue sizes are also correct and dependent on the controller's power. An increase in controller power reduces the average queue size.

Our study shows the advantages of using the  $PI^\alpha$  controller as an AQM mechanism. The use of the Fractional Calculus in the proposed mechanism allows for more precise regulation of the queue length in the communication node. The  $PI^\alpha$  controller used as the AQM mechanism requires more computing power and more memory than the RED algorithm. However, the computational cost of the proposed mechanism becomes less important with increasing routers computing power.

The results presented here also confirm the correctness of the model introduced in our previous article [32]. A single diffusion experiment takes several minutes. This is a disadvantage compared to other

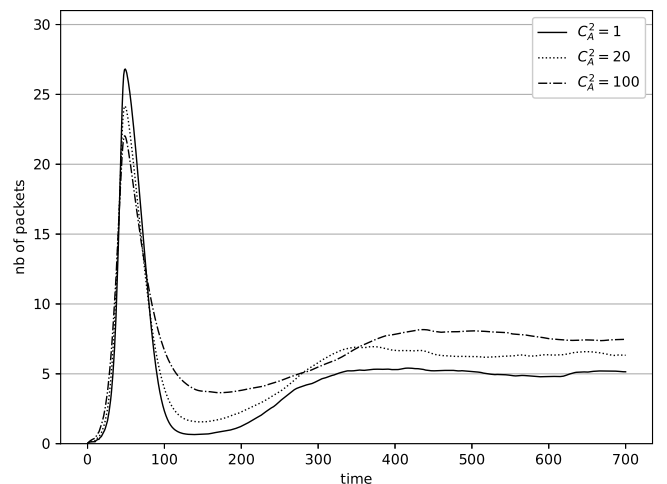


Fig. 7. The router mean queue length,  $G/M/1/30$  queue, impact of traffic variation, 1st set of  $PI^\alpha$  parameters.

methods (e.g. fluid-flow approximation [68]). But unlike this method, diffusion approximation offers more accurate results.

In our future work, we will focus on the diffusion models reflecting real Internet traffic (increased number of transmitters and the presence of both TCP and UDP streams).

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

This research was financed by the National Science Center, Poland project no. 2017/27/B/ST6/00145.

## References

- [1] S. Floyd, V. Jacobson, Random early detection gateways for congestion avoidance, *IEEE/ACM Trans. Netw.* 1 (4) (1993) 397–413, <http://dx.doi.org/10.1109/90.251892>.
- [2] W. Chang Feng, D. Kandlur, D. Saha, Adaptive packet marking for maintaining end to end throughput in a differentiated Service Internet, *IEEE/ACM Trans. Netw.* 7 (5) (1999) 685–697.
- [3] M. May, T. Bonald, J. Bolot, Analytic evaluation of RED performance, in: *Proceedings of the IEEE Infocom, Tel-Aviv, Izrael, 2000*.
- [4] Z.-H. Li, Y. Liu, Y.-W. Jing, Active queue management algorithm for TCP networks with integral backstepping and minimax, *Int. J. Control Autom. Syst.* 17 (4) (2019) 1059–1066, <http://dx.doi.org/10.1007/s12555-018-0447-5>.
- [5] J. Domańska, A. Domański, T. Czachórski, Fluid flow analysis of RED algorithm with modified weighted moving average, in: *Modern Probabilistic Methods for Analysis of Telecommunication Networks*, Vol. 356, Springer Berlin Heidelberg, 2013, pp. 50–58, [http://dx.doi.org/10.1007/978-3-642-35980-4\\_7](http://dx.doi.org/10.1007/978-3-642-35980-4_7).
- [6] J. Domańska, A. Domański, T. Czachórski, M. Pagano, The fluid flow approximation of the TCP vegas and reno congestion control mechanism, in: *Computer and Information Sciences*, Springer International Publishing, 2016, pp. 193–200, [http://dx.doi.org/10.1007/978-3-319-47217-1\\_21](http://dx.doi.org/10.1007/978-3-319-47217-1_21).
- [7] Y. Mishura, Diffusion approximation of recurrent schemes for financial markets, with application to the Ornstein–Uhlenbeck process, *Opuscula Math.* 35 (2015) 99, <http://dx.doi.org/10.7494/OpMath.2015.35.1.99>.
- [8] G. Newell, Applications of queueing theory, *J. Appl. Math. Mech.* 53 (3) (1971) <http://dx.doi.org/10.1002/zamm.19730530310>.
- [9] M. Snover, S. Heppell, Application of diffusion approximation for risk assessments of sea turtle populations, in: *Ecological Applications: a Publication of the Ecological Society of America*, Vol. 19, 2009, pp. 774–785, <http://dx.doi.org/10.1890/07-1111.1>.
- [10] T. Czachórski, F. Pekergin, Diffusion approximation as a modelling tool, in: *Network Performance Engineering. A Handbook on Convergent Multi-Service Networks and Next Generation Internet*, Springer Berlin Heidelberg, 2011, pp. 447–476, [http://dx.doi.org/10.1007/978-3-642-02742-0\\_20](http://dx.doi.org/10.1007/978-3-642-02742-0_20).
- [11] Y. Nonaka, S. Nogami, Evaluation of diffusion approximation for the G/G/1 queueing model, in: *8th Asia-Pacific Symposium on Information and Telecommunication Technologies*, 2010, pp. 1–6.
- [12] T. Czachórski, T. Nycz, F. Pekergin, Priority disciplines - A diffusion approach, in: *23rd International Symposium on Computer and Information Sciences, ISCIS, 2008*, pp. 1–4, <http://dx.doi.org/10.1109/ISCIS.2008.4717959>.
- [13] T. Czachórski, T. Nycz, F. Pekergin, Transient states of priority queues - A diffusion approximation study, in: *Fifth Advanced International Conference on Telecommunications*, 2009, pp. 44–51, <http://dx.doi.org/10.1109/AICT.2009.14>.
- [14] T. Czachórski, K. Grochla, F. Pekergin, Stability and dynamics of TCP-NCR(DCR) protocol in presence of UDP flows, in: *Wireless Systems and Mobility in Next Generation Internet*, Springer Berlin Heidelberg, 2007, pp. 241–254, [http://dx.doi.org/10.1007/978-3-540-70969-5\\_18](http://dx.doi.org/10.1007/978-3-540-70969-5_18).
- [15] J. Domańska, A. Domański, Czachórski, Comparison of AQM control systems with the use of fluid flow approximation, *Commun. Comput. Inf. Sci.* (2012) 82–90, [http://dx.doi.org/10.1007/978-3-642-31217-5\\_9](http://dx.doi.org/10.1007/978-3-642-31217-5_9).
- [16] D. Towsley, W. Gong, K. Hollot, Y. Liu, V. Misra, Fluid methods for modeling large, heterogeneous networks, *NTIS* (2005).
- [17] T. Czachórski, A method to solve diffusion equation with instantaneous return processes acting as boundary conditions, *Bull. Pol. Acad. Sci. Tech. Sci.* 41 (4) (1993) 417–451.
- [18] S. Halfin, W. Whitt, Heavy-traffic limits for queues with many exponential servers, *Oper. Res.* 29 (1981) 567–588, <http://dx.doi.org/10.1287/opre.29.3.567>.
- [19] M. Hassan, R. Jain, *High Performance TCP/IP Networking - Concepts, Issues and Solutions*, Pearson Education Inc, 2004.
- [20] B. Zheng, M. Atiquzzaman, DSRED: A new queue management scheme for next generation networks, in: *The 25th Annual IEEE Conference on Local Computer Networks*, 2000, pp. 242–251.
- [21] B. Zheng, M. Atiquzzaman, Improving performance of active queue management over heterogeneous networks, in: *ICC 2001: International Conference on Communications*, Vol. 8, 2001, pp. 2375–2379, <http://dx.doi.org/10.1109/ICC.2001.936557>.
- [22] K. Zhou, K. Yeung, V. Li, Nonlinear RED: A simple yet efficient active queue management scheme, *Comput. Netw.: Int. J. Comput. Telecommun. Netw.* 50 (18) (2006) 3784–3794, <http://dx.doi.org/10.1016/j.comnet.2006.04.007>.
- [23] J. Domańska, A. Domański, D. Augustyn, J. Klamka, A RED modified weighted moving average for soft real-time application, *Int. J. Appl. Math. Comput. Sci.* 24 (3) (2014) 697–707, <http://dx.doi.org/10.2478/amcs-2014-0051>.
- [24] C. Hollot, V. Misra, D. Towsley, W. Gong, On designing improved controllers for AQM routers supporting TCP flows, in: *Twentieth Annual Joint Conference of the IEEE Computer and Communications Society. Proceedings IEEE INFOCOM 2001*, Vol. 3, 2001, pp. 1726–1734, <http://dx.doi.org/10.1109/INFCOM.2001.916670>.
- [25] P. Quet, H. Ozbay, On the design of AQM supporting TCP flows using robust control theory, *IEEE Trans. Autom. Control* Vol. 49 (6) (2004) 1031–1036, <http://dx.doi.org/10.1109/TAC.2004.829643>.
- [26] W. Michiels, D. Melchor-Aquilar, S. Niculescu, Stability analysis of some classes of TCP/AQM networks, *Int. J. Control* 15 (2006) 1–12, <http://dx.doi.org/10.1080/00207170600810701>, Manuscript.
- [27] D. Melchor-Aquilar, V. Castillo-Tores, Stability analysis of proportional-integral AQM controllers supporting TCP flows, *Comput. Sistemas* 10 (2007) 401–414, [http://dx.doi.org/10.1016/S1474-6670\(17\)69284-X](http://dx.doi.org/10.1016/S1474-6670(17)69284-X).
- [28] D. Ustebay, H. Ozbay, Switching resilient PI controllers for active queue management of TCP flows, in: *Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control*, 2007, pp. 574–578, <http://dx.doi.org/10.1109/ICNSC.2007.372842>.
- [29] D. Melchor-Aquilar, S. Niculescu, Computing non-fragile PI controllers for delay models of TCP/AQM networks, *Int. J. Control* 82 (12) (2009) 2249–2259, <http://dx.doi.org/10.1080/00207170902984741>.
- [30] W. Krajewski, U. Viaro, On robust fractional order PI controller for TCP packet flow, in: *BOS Conference: Systems and Operational Research*, Warsaw, Polan, 2014.
- [31] R. Laalaoua, T. Atmaca, S. Jędrús, T. Czachórski, Diffusion model of RED control mechanism, in: *Networking*, Springer Berlin Heidelberg, 2001, pp. 107–116, [http://dx.doi.org/10.1007/3-540-47728-4\\_11](http://dx.doi.org/10.1007/3-540-47728-4_11).
- [32] A. Domański, J. Domańska, T. Czachórski, J. Klamka, J. Szyguła, D. Marek, Diffusion approximation model of TCP newreno congestion control mechanism, *Springer Nat. Comput. Sci.* 1 (1) (2019) 43, <http://dx.doi.org/10.1007/s42979-019-0032-x>.
- [33] G.F. Newell, Queues with time-dependent arrival rates. I — The transition through saturation, *J. Appl. Probab.* 2 (2) (1968) 436–451, <http://dx.doi.org/10.2307/3212264>.
- [34] G.F. Newell, Queues with time-dependent arrival rates. II — The maximum queue and the return to equilibrium, *J. Appl. Probab.* 2 (3) (1968) 579–590, <http://dx.doi.org/10.2307/3211923>.
- [35] G.F. Newell, Queues with time-dependent arrival rates. III — A mild rush hour, *J. Appl. Probab.* 2 (3) (1968) 591–606, <http://dx.doi.org/10.2307/3211924>.
- [36] G.F. Newell, Applications of Queueing Theory, in: *Monographs on Statistics and Applied Probability*, Chapman and Hall, London, 1971, <http://dx.doi.org/10.1002/zamm.19730530310>.
- [37] H. Kobayashi, Application of the diffusion approximation to queueing networks I: Equilibrium queue distributions, *J. ACM (JACM)* 21 (2) (1974) 316–328, <http://dx.doi.org/10.1145/321812.321827>.
- [38] E. Gelenbe, On approximate computer systems models, *J. ACM* 22 (2) (1975) 261–269, <http://dx.doi.org/10.1145/321879.321888>.
- [39] D. Iglehart, W. Whitt, Multiple channel queues in heavy traffic - part I, *Adv. Appl. Probab.* 2 (1) (1970) 150–177, <http://dx.doi.org/10.2307/3518347>.
- [40] D. Iglehart, W. Whitt, Multiple channel queues in heavy traffic. Part II: Sequences, networks, and batches, *Adv. Appl. Probab.* 2 (2) (1970) 355–369, <http://dx.doi.org/10.2307/1426324>.
- [41] W. Whitt, Multiple channel queues in heavy traffic - part III: Random server selection, *Adv. Appl. Probab.* 2 (2) (1970) 370–375, <http://dx.doi.org/10.2307/1426325>.
- [42] D. Iglehart, Weak convergence in queueing theory, *Adv. Appl. Probab.* 5 (3) (1973) 570–594, <http://dx.doi.org/10.2307/1425835>.
- [43] H. Kobayashi, *Modelling and Analysis: An Introduction to System Performance Evaluation Methodology*, Addison-Wesley, Reading, Massachusetts, 1978.
- [44] R. Cox, H.D. Miller, *The Theory of Stochastic Processes*, Chapman and Hall, London, 1965.
- [45] H. Stehfest, Algorithm 368: Numeric inversion of Laplace transform, *Commun. ACM* 13 (1) (1970) 47–49, <http://dx.doi.org/10.1145/361953.361969>.
- [46] T. Czachórski, T. Nycz, F. Pekergin, Transient states analysis - diffusion approximation as an alternative to Markov models, fluid-flow approximation and simulation, in: *IEEE Symposium on Computers and Communications*, 2009, pp. 13–18.
- [47] V. Morales-Delgado, J. Gómez-Aguilar, M. Taneco-Hernandez, Analytical solution of the time fractional diffusion equation and fractional convection-diffusion equation, *Rev. Mex. Fis.* 65 (1) (2019) 82–88, <http://dx.doi.org/10.31349/RevMexFis.65.82>.
- [48] J. Gómez-Aguilar, H. Yezpez-Martinez, R. Escobar Jimenez, V. Olivares Peregrino, J. Reyes, I. Sosa, Series solution for the time-fractional coupled mKdV equation using the homotopy analysis method, *Math. Probl. Eng.* (2016) 1–21, <http://dx.doi.org/10.1155/2016/7047126>.

- [49] A. Atangana, J. Gómez-Aguilar, Decolonisation of fractional calculus rules: Breaking commutativity and associativity to capture more natural phenomena, *Eur. Phys. J. Plus* 133 (2018) <http://dx.doi.org/10.1140/epjp/i2018-12021-3>.
- [50] A. Atangana, J. Gómez-Aguilar, Fractional derivatives with no-index law property: Application to chaos and statistics, *Chaos Solitons Fractals* 114 (2018) 516–535, <http://dx.doi.org/10.1016/j.chaos.2018.07.033>.
- [51] T. Abdeljawad, On Riemann and Caputo fractional differences, *Comput. Math. Appl.* 62 (3) (2011) 1602–1611, <http://dx.doi.org/10.1016/j.camwa.2011.03.036>.
- [52] T. Abdeljawad, D. Baleanu, F. Jarad, R.P. Agarwal, Fractional sums and differences with binomial coefficients, in: *Discrete Dynamics in Nature and Society*, Hindawi Publishing Corporation, 2013, <http://dx.doi.org/10.1155/2013/104173>.
- [53] K. Miller, B. Ross, *An Introduction to the Fractional Calculus and Fractional Differential Equations*, Wiley, New York, 1993.
- [54] T. Kaczorek, Reachability and controllability to zero of positive fractional discrete-time systems, in: *European Control Conference, ECC, 2007*, pp. 1708–1712.
- [55] I. Podlubny, Fractional order systems and  $PI^{\alpha}D^{\beta}$  controllers, *IEEE Trans. Autom. Control* 44 (1) (1999) 208–214.
- [56] Y. Chen, I. Petras, D. Xue, Fractional order control - A tutorial, in: *American Control Conference, 2009*, pp. 1397–1411.
- [57] Y. Luo, Y. Chen, Fractional-order proportional derivative controller for a class of fractional order systems, *Automatica* 45 (10) (2009) 2446–2450.
- [58] K. Oprzdkiewicz, M. Podsiado, The fractional order PID control of the forced air heating system, *Pomiary Autom. Robotyka* 23 (2019) 5–10, [http://dx.doi.org/10.14313/PAR\\_231/5](http://dx.doi.org/10.14313/PAR_231/5).
- [59] K. Bingi, R. Ibrahim, M. Karsiti, S. Hassan, Frequency response based curve fitting approximation of fractional-order PID controllers, *Int. J. Appl. Math. Comput. Sci.* 29 (2019) 311–326, <http://dx.doi.org/10.2478/amcs-2019-0023>.
- [60] J. Domańska, A. Domański, T. Czachórski, J. Klamka, Self-similarity traffic and AQM mechanism based on non-integer order  $PI^{\alpha}D^{\beta}$  controller, in: *Communications in Computer and Information Science*, Vol. 718, Springer International Publishing, 2017, pp. 336–350, [http://dx.doi.org/10.1007/978-3-319-59767-6\\_27](http://dx.doi.org/10.1007/978-3-319-59767-6_27).
- [61] J. Domańska, A. Domański, T. Czachórski, J. Klamka, The use of a non-integer order PI controller with an Active Queue Management Mechanism, *Int. J. Appl. Math. Comput. Sci.* 26 (2016) 777–789, <http://dx.doi.org/10.1515/amcs-2016-0055>.
- [62] J. Domańska, A. Domański, T. Czachórski, J. Klamka, J. Szyguła, The AQM Dropping Packet Probability Function Based on Non-integer Order  $PI^{\alpha}D^{\beta}$  Controller, in: *Lecture Notes in Electrical Engineering*, Vol. 496, Springer International Publishing, 2019, pp. 36–48, [http://dx.doi.org/10.1007/978-3-319-78458-8\\_4](http://dx.doi.org/10.1007/978-3-319-78458-8_4).
- [63] J. Domańska, A. Domański, T. Czachórski, J. Klamka, D. Marek, J. Szyguła, GPU accelerated non-integer order  $PI^{\alpha}D^{\beta}$  controller used as AQM mechanism, in: *Communications in Computer and Information Science*, Vol. 860, Springer Verlag, Berlin, 2018, pp. 286–299, [http://dx.doi.org/10.1007/978-3-319-92459-5\\_23](http://dx.doi.org/10.1007/978-3-319-92459-5_23).
- [64] M. Ciesielski, J. Leszczynski, A numerical method for solution of ordinary differential equations of fractional order, in: *Parallel Processing and Applied Mathematics*, Vol. 2328, Springer Berlin Heidelberg, 2002, pp. 695–702, [http://dx.doi.org/10.1007/3-540-48086-2\\_77](http://dx.doi.org/10.1007/3-540-48086-2_77).
- [65] I. Podlubny, *Fractional Differential Equations*, Vol. 198, Academic Press, San Diego, USA, 1999.
- [66] J. Domańska, A. Domański, T. Czachórski, J. Klamka, D. Marek, J. Szyguła, The influence of the traffic self-similarity on the choice of the non-integer order  $PI^{\alpha}$  controller parameters, in: *Communications in Computer and Information Science*, Vol. 935, Springer International Publishing, 2018, pp. 76–83, [http://dx.doi.org/10.1007/978-3-030-00840-6\\_9](http://dx.doi.org/10.1007/978-3-030-00840-6_9).
- [67] J. Domańska, A. Domański, T. Czachórski, J. Klamka, J. Szyguła, D. Marek, AQM mechanism with the dropping packet function based on the answer of several  $PI^{\alpha}$  controllers, in: *Communications in Computer and Information Science*, Springer Verlag, Berlin, 2019, [http://dx.doi.org/10.1007/978-3-030-21952-9\\_29](http://dx.doi.org/10.1007/978-3-030-21952-9_29).
- [68] J. Domańska, A. Domański, T. Czachórski, J. Klamka, Fluid flow approximation of time-limited TCP/UDP/XCP streams, *Bull. Pol. Acad. Sci.: Tech. Sci.* 62 (2) (2014) 217–225, <http://dx.doi.org/10.2478/bpasts-2014-0021>.



**Dariusz Marek**, Ph.D. Student in the Department of Distributed Systems and Informatic Devices, Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology. His main research interests are related to algorithms of Active Queue Management in Computer Networks.



**Adam Domański**, Ph.D., Eng., Professor of the Silesian University of Technology, works in Computer Equipment Group of the Department of Distributed Systems and Informatic Devices, Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology. His main research interest in the computer networks domain is congestion control in packet networks.



**Joanna Domańska**, Ph.D., Eng., Institute Professor, has been a member of the Computer Systems Modelling and Performance Evaluation Group of the Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice, Poland, since 1994 where she has been serving as the Leader since 2018. Her main areas of research interest include performance modeling methods for computer networks, particularly the modeling of network traffic intensity and the quality of service (QoS) related problems.



**Tadeusz Czachórski**, Ph.D., Prof., the head of the Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences. His main areas of interest are mathematical and numerical methods for modeling and evaluation of computer networks.



**Jerzy Klamka**, Ph.D., Prof., a full member of the Polish Academy of Sciences, works in the Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences. His main areas of research are controllability and observability of linear and nonlinear dynamical systems, and mathematical foundations of quantum computations. He is an author of monographs and numerous papers published in international journals.



**Jakub Szyguła**, Ph.D. Student in the Department of Distributed Systems and Informatic Devices, Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology. His main research interests are related to algorithms of Active Queue Management in Computer Networks.

## **Rozdział 8**

# **Diffusion approximation model of TCP NewReno congestion control mechanism**



# Diffusion Approximation Model of TCP NewReno Congestion Control Mechanism

Adam Domański<sup>1</sup> · Joanna Domańska<sup>2</sup> · Tadeusz Czachórski<sup>2</sup> · Jerzy Klamka<sup>2</sup> · Jakub Szyguła<sup>1</sup> · Dariusz Marek<sup>1</sup>

Received: 22 July 2019 / Accepted: 2 September 2019  
© The Author(s) 2019

## Abstract

The aim of the article is to contribute to the modeling and evaluation of TCP/IP traffic control in case of active queue management mechanisms implemented in congestion router. We consider mechanisms based on RED and fractional-order  $PI^\alpha$  controller. A new model based on diffusion approximation is developed and its results are compared with the results of an existing fluid flow model. Both models are used to investigate the influence of AQM control parameters on the flow control. Their numerical results concerning dynamics of congestion window, loss probability, and traffic changes are quite similar. The advantage of the new model lies in giving more detailed information: the distribution of router queue and the distribution of introduced delay, not only their mean values as it is in case of fluid flow model. The variability of transmission time (jitter) is a factor of transmission quality, and therefore, the proposed model may be useful in further studies on TCP/IP quality of service.

**Keywords** Diffusion approximation · AQM · Non-integer order  $PI^\alpha$  controller · G/G/1/N queueing model · Fluid flow approximation · Congestion control

## Introduction

In the article, we present the diffusion model of the TCP NewReno transmission control where the congestion router implements an Active Queue Mechanism, based either on

classical RED or fractional-order  $PI^\alpha$  controller. The AQM algorithms, recommended by IETF, enhance the efficiency of transfers and cooperate with TCP congestion window mechanism in adapting the intensity of flows to the congestion at a network [2]. They incorporate mechanisms of preventive packet dropping performed when there is still a place to store packets in the queue, to advertise that the queue is increasing and the danger of congestion is approaching. For most algorithms of this type, the packets are rejected randomly and the probability of packet dropping increases with the queue length. The basic active queue management algorithm is Random Early Detection (RED) algorithm proposed in 1993 [14]. Since then, numerous modifications to this algorithm have been discussed [22, 23, 41]. They differ in the way how the probability of the loss of packets is determined, a recently investigated option is to use in this purpose a  $PI^\alpha$  controller [24].

## Our Contribution

The behavior of two AQM mechanisms applied to control Internet traffic is investigated. We compare in TCP transport protocol the basic RED algorithm and mechanism using a fractional-order controller. We explore the influence of parameters of the  $PI^\alpha$  controller on the queue

---

This article is part of the topical collection “Modelling methods in Computer Systems, Networks and Bioinformatics” guest edited by Erol Gelenbe.

---

✉ Adam Domański  
Adam.Domanski@polsl.pl

Joanna Domańska  
joanna@iitis.pl

Tadeusz Czachórski  
tadek@iitis.pl

Jerzy Klamka  
jklamka@iitis.pl

Jakub Szyguła  
Jakub.Szygula@polsl.pl

Dariusz Marek  
Dariusz.Marek@polsl.pl

<sup>1</sup> Institute of Informatics, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland

<sup>2</sup> Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland

length and evolution of the TCP congestion window. The second aim is to investigate the diffusion approximation perspectives for modeling TCP window changes.

A popular method for assessing the evolution of TCP congestion window is fluid flow approximation [29]. Therefore, we compare the proposed diffusion model with the other based on fluid flow approach. We are interested in diffusion model, because it gives also the distribution of the transport time, hence the information on the transmission jitter.

The remainder of the paper is organized as follows: section “[Related works](#)” describes the works related to this article. Section “RED and Non-integer order  $PI^\alpha$  controller” presents bases of RED and  $PI^\alpha$  controller. The fluid flow model is presented in the section “[Fluid Flow model](#)” and the basic notions of the diffusion approximation are presented in the section “[Diffusion approximation](#)”. Section “[Diffusion approximation models of TCP NewReno mechanism](#)” investigates the diffusion approximation model of TCP NewReno mechanism. Concluding remarks are presented in section “[Conclusion](#)”.

## Related Works

The Active Queue Management mechanism is based on preventive packet dropping when there is still a place to store packets, the probability of packet rejection is increasing together with the level of congestion. Packets are dropped randomly, and hence, only chosen users are notified and the global synchronization of connections is avoided. A detailed discussion of the active queue management goals may be found in [2]. The AQM mechanism may be seen as part of a closed loop control of TCP/IP traffic intensity. The feedback control system includes the sender’s TCP congestion window increasing or decreasing the TCP flow as a function of losses in the queue of packets at a bottleneck router reacting to the changes in the input flow. AQM determines the loss probability which, in turn, after a certain delay, modifies the congestion window. Fluid flow modeling techniques were proposed to model the behavior of Internet traffic, e.g., [18, 19, 25], and they proved to be suitable for TCP traffic modeling [37]. Existing fluid flow models can simulate persistent TCP flows [31] and time-limited TCP flows [5]. The short-lived TCP flow, UDP flow, and XCP models have been presented in [1, 36]. Fluid network models with heterogenous flows were rarely investigated [31, 40]. In the last years, various AQM algorithms were proposed and studied [20, 21, 33]. Most of these studies were performed in a simulation environment. The analytical studies of AQM models are

computationally complex, and hence, usually, they were investigated in open-loop scenario [4, 12, 26].

## RED and Non-integer Order $PI^\alpha$ Controller

The traffic control at TCP/IP is a classical closed loop control where the input signal, i.e., traffic intensity  $\lambda$  being sent is determined as a function of loss probability  $p_{\text{loss}}$  witnessed at the queue of the congestion router between the sender and receiver. When  $p_{\text{loss}} = 0$ , the traffic intensity is increased, and if  $p_{\text{loss}} > 0$ , traffic intensity is decreased. With passive router queue losses start when the queue is full, with AQM, they start earlier anticipating the total congestion of the router.

In particular, RED algorithm determines dropping probability as a function of the weighted moving average  $\text{avg}$  acting as a low-pass filter and computed at the arrival of every packet; for the packet  $i$ :

$$\text{avg}_i = (1 - w)\text{avg}_{i-1} + wq_i,$$

where  $q_i$  is the queue seen by packet  $i$  and  $\text{avg}_{i-1}$  is the moving average computed at arrival of previous packet. The loss probability for small values of  $\text{avg}$  is  $p_{\text{RED}} = 0$ , and then, it is increasing linearly from 0 to  $p_{\text{max}}$  between two thresholds  $\text{Min}_{\text{th}}$  and  $\text{Max}_{\text{th}}$ :

$$P_{\text{RED}}(\text{avg}) = p_{\text{max}} \frac{\text{avg} - \text{Min}_{\text{th}}}{\text{Max}_{\text{th}} - \text{Min}_{\text{th}}} \quad (1)$$

and becomes  $p_{\text{RED}} = 1$  for  $\text{avg} > \text{Max}_{\text{th}}$ . Probability  $p_{\text{loss}}$  may be also determined by a controller having as the input the “error”, i.e., the difference between the current queue  $q_i$  and a specified desired queue  $q_0$ . This controller may be a classical PID one or, as it is investigated recently, a proportional-integral,  $PI^\alpha$  fractional non-integer order controller.

Fractional-Order Derivatives and Integrals (FOD/FOI) are a natural extension of the well-known integrals and derivatives. Differ-integrals of non-integer orders enable better and more precise control of physical processes [35].

In case of  $PI^\alpha$  controller, the loss probability  $p_i$  of a packet  $i$  is:

$$p_i = \max\{0, -(K_p e_i + K_I \Delta^\alpha e_i)\}, \quad (2)$$

and it depends on the error  $e_i = q_i - q_0$ , the proportional and integral terms  $K_p, K_I$ , and the integration order  $\alpha$ . The articles [6–8, 10] discuss their impact.

Packet drop probabilities are determined at discrete moments of packet arrivals; there is only one definition of discrete differ-integrals of non-integer order. This definition [28] is a generalization of the traditional definition of the difference of integer order to the non-integer order, and it is analogous to a generalization used in Grunwald–Letnikov (GrLET) formula [3, 34].

For a given sequence  $f_0, f_1, \dots, f_j, \dots, f_k$ :

$$\Delta^\alpha f_k = \sum_{j=0}^k (-1)^j \binom{\alpha}{j} f_{k-j}, \tag{3}$$

where  $\alpha \in R$  is generally a non-integer fractional order,  $f_k$  is a differentiated discrete function, and  $\binom{\alpha}{j}$  is generalized Newton symbol defined as follows:

$$\binom{\alpha}{j} = \begin{cases} 1 & \text{for } j = 0 \\ \frac{\alpha(\alpha - 1)(\alpha - 2)\dots(\alpha - j + 1)}{j!} & \text{for } j = 1, 2, \dots \end{cases} \tag{4}$$

### Fluid Flow Model

Fluid flow model presented in [29] demonstrates TCP protocol dynamics. The model ignores the TCP timeout mechanisms and is based on the following nonlinear differential equation describing the evolution of the congestion window size [22]:

$$\frac{dW_i(t)}{dt} = \frac{1}{R_i(t)} - \frac{W_i(t)}{2} \frac{W_i(t - R(t))}{R_i(t - R_i(t))} p(t - R_i(t)). \tag{5}$$

The next equation is related to the queue evolution of the congested router:

$$\frac{dq(t)}{dt} = \sum_{i=1}^N \frac{W_i(t)}{R_i(t)} - C, \tag{6}$$

where  $W_i$  is the expected TCP congestion window size (expressed in packets) for a flow  $i$ , it defines how many packets may be sent without waiting for the acknowledgements of previous packets reception,  $R_i$  is the round-trip time,  $R_i = q/C + T_p$ , the sum  $\sum \frac{W_i}{R_i}$  is the total input flow to the congestion router,  $q$  is the queue length in packets,  $C$  is the link capacity (packets/time unit), the constant output flow of the router,  $T_p$  is the propagation delay,  $N$  is the number of TCP sessions passing through the router, and  $p$  is the packet drop probability.

Equations (5) and (6) should be supplemented with additional one giving dropping probability, and it may be Eq. (1) or Eq. (2).

The maximum values of  $q$  and  $W$  depend on the buffer capacity and maximum window size. This fluid flow model may be extended to any network topology with any number of TCP flows [39]. The article [32] presents a numerical example for a real network topology with 134 023 nodes and 50 000 flows.

### Diffusion Approximation

The method of diffusion approximation, e.g., [15–17, 30] replaces the number of customers in a queueing system  $N(t)$  which is a discrete value process by a continuous diffusion process  $X(t)$ . The density function  $f(x, t; x_0)$ :

$$f(x, t; x_0) dx = P[x \leq X(t) < x + dx | X(0) = x_0] \tag{7}$$

of  $X(t)$ , determined by the diffusion equation:

$$\frac{\partial f(x, t; x_0)}{\partial t} = \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x} \tag{8}$$

approximates the distribution of the number of customers in the system. Both processes,  $N(t)$  and  $X(t)$ , have normal distribution of changes. In case of  $N(t)$ , for a G/G/1 queueing system where  $A(x)$  and  $B(x)$  are the interarrival and service time distribution station, the mean and the variance of the changes observed in  $\Delta t$  are  $(\lambda - \mu)\Delta t$  and  $(\sigma_A^2 \lambda^3 + \sigma_B^2 \mu^3)\Delta t$ , where  $E[A] = 1/\lambda$ ,  $E[B] = 1/\mu$ ,  $Var[A] = \sigma_A^2$ ,  $Var[B] = \sigma_B^2$ .

For the diffusion process, the mean and the variance of the changes in  $dt$  are  $\beta dt$  and  $\alpha dt$ , where  $\alpha$  and  $\beta$  are the parameters of the diffusion equation. If we assume  $\beta = \lambda - \mu$ ,  $\alpha = \lambda^3 \sigma_A^2 + \mu^3 \sigma_B^2$ , both the processes have not only normally distributed changes, but the mean and the variance of these distributions depend in the same way on the observation time. The process  $N(t)$  is non-negative, and thus, also  $X(t)$  should be limited to  $x \geq 0$  by a barrier at  $x = 0$ . In case of a limited queue, e.g., in G/G/1/L station, the second barrier is introduced at  $x = L$  to limit the diffusion process to the interval  $[0, L]$ . As it was proposed by Gelenbe [15], we apply here the absorbing barriers with instantaneous jumps: when the process comes to  $x = 0$ , it is absorbed by the barrier, and then, after a certain time, jumps to  $x = 1$ . Similarly, if  $X(t)$  comes to  $x = L$ , it is absorbed and then jumps to  $x = L - 1$ . The time spent in barriers corresponds to the idle time or to saturation time of the system. The diffusion equation with two barriers has the form:

$$\begin{aligned} \frac{\partial f(x, t; x_0)}{\partial t} &= \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x} \\ &\quad + \lambda p_0(t) \delta(x - 1) + \lambda p_L(t) \delta(x - L + 1), \\ \frac{dp_0(t)}{dt} &= \lim_{x \rightarrow 0} \left[ \frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0) \right] - \lambda p_0(t), \\ \frac{dp_L(t)}{dt} &= \lim_{x \rightarrow L} \left[ \frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0) \right] - \lambda p_L(t), \end{aligned} \tag{9}$$

where:  $p_0(t) = P[X(t) = 0]$ ,  $p_L(t) = P[X(t) = L]$ .

Of course, in case of G/G/1, model terms related to  $p_L$  are omitted, as there is only one barrier at  $x = 0$ . The steady-state solution of Eq. (9) is relatively straightforward and has the form of exponential functions, different in intervals  $[0, 1]$ ,  $[1, L - 1]$ ,  $[L - 1, L]$  with  $p_0$  and  $p_L$  given by normalization, see [15]. In

case of transient states, as applied here, we use a solution in the form of Laplace transform presented in [38] which is then inverted numerically. The solution of the diffusion process with jumps from the barriers is expressed as a convolution of solutions obtained for purely absorbing barriers and the intensity of jumps is given by balance equations for flows coming and leaving the barriers.

These G/G/1 and G/G/1/L models are valid for transient behavior, but assume constant parameters  $\alpha, \beta$ . To include time-varying model parameters (e.g., for  $\lambda = \lambda(t)$ ), we should divide the time axis into short periods where the parameters of the model are constant. In this approach, the results obtained at the end of a period serve as the initial condition for the next period.

## Diffusion Approximation Models of TCP NewReno Mechanism

Here, we propose a diffusion approximation model of the TCP NewReno connection having an intermediate router with AQM ( $PI^\alpha$  or RED) policy. Diffusion approximation gives the router queue distribution at time  $t$ . Its mean value modifies the probability of packet rejection. This probability changes the input stream intensity  $\lambda$ , as the congestion window is increased by one in the case of lossless transmission or is halved in the case of a packet loss.

As in fluid flow model, we assume that the flow  $\lambda$ , the size of the congestion window  $W$ , and the delay  $q/\mu$  of the router are linked together in the following way:

$$\lambda = \frac{W\mu}{q}, \quad (10)$$

where  $1/\mu$  is mean service time. The diffusion model is performed calculations are carried out inside intervals of the length  $\Delta t = 1/\lambda$ . As  $\lambda$  is changing, the length of the interval is also changing. We obtain a queue distribution at time  $t_i$  at the end of the interval  $i$ . Based on it, we have the average queue length  $E[q_i]$  and then the probability of packet rejection  $p_i$  which due to AQM algorithm defines new value of  $\lambda$ ,  $\lambda_{i+1} = \lambda_i + \Delta\lambda_i$ , where

$$\Delta\lambda_i = \frac{\mu_i}{E[q_i]} - \frac{\lambda_i^2}{2} \frac{E[q_i]}{\mu} p_i. \quad (11)$$

In the time  $t_{i+1} = t_i + 1/\lambda_i$ , we repeat the calculations for new value of  $\lambda$ .

The changes to  $\Delta\lambda$  in the time  $t$  depend on the source intensity, dispatching time, and queue length.

In numerical examples, we use the G/G/1/30 model, and the length of the buffer  $L = 30$  is enough to neglect full congestion probability. We only consider losses due to the AQM mechanism. Below, we compare the results obtained by the proposed solution with the results obtained by the fluid flow method. Both models give similar results. The advantage of the proposed diffusion model compared to fluid flow one is the ability to obtain more detailed results concerning transmission time. In this new model, we have the distribution of the queue, and in fluid flow approximation, we have only its mean value. The distribution of the queue size  $f(n, t, n_0)$  allows us to compute the density function  $r(t, x)$  of the router delay:

$$r(t, x) = \sum_{n=0}^L f(n, r; n_0) b(x)^{(n+1)*}, \quad (12)$$

where  $i^*$  denotes the  $i$ -fold convolution and  $b(x)$  is probability density function of the service time distribution, and this way the information on the jitter of the transmission time.

Figures 1, 2, 3 refer to  $PI^\alpha$  controller display the behavior of the AQM queue due to changes of the congestion window and the evolution of congestion  $\rho = \lambda/\mu$  itself for selected sets of parameters  $K_p, K_f, \alpha$ . They visualize the similarity of results obtained by fluid flow and diffusion approximations. Figure 1 displays the system behavior in the case of a “strong”, i.e., giving relatively high loss probability and low queue length. In this case, the queue size tends to ten packets. The controller in Fig. 2 is weaker. In its steady state, the mean size of the queue is around 11 packets. Figure 3 presents the influence of  $\alpha$  on the queue length. The queue decreases as the integral order increases. The same may be observed for the RED algorithm. The queue size decreases as the parameter  $p_{max}$  increases (Fig. 4). Figure 5 illustrates the influence of  $p_{max}$  in RED.

Tables 1, 2 show the obtained average queue lengths. The detailed results also show small differences between the results obtained with both methods.

The results of the  $PI^\alpha$  controller presented in Table 1 show the influence of the parameter  $PI^\alpha$  on the average queue length. Increasing the integration order decreases the average queue size. The mean queue length depends on the integral order and oscillates between 10 and 18 packets. It is associated with a significant increase in the probability of a packet loss. The same relationship is clearly visible both for the fluid flow method and diffusion G/G/1/L queue. Table 2 presents the results of the for the RED mechanism. The increasing of  $P_{max}$  results in a reduction of the average queue length. The reasons for these results are the same as in



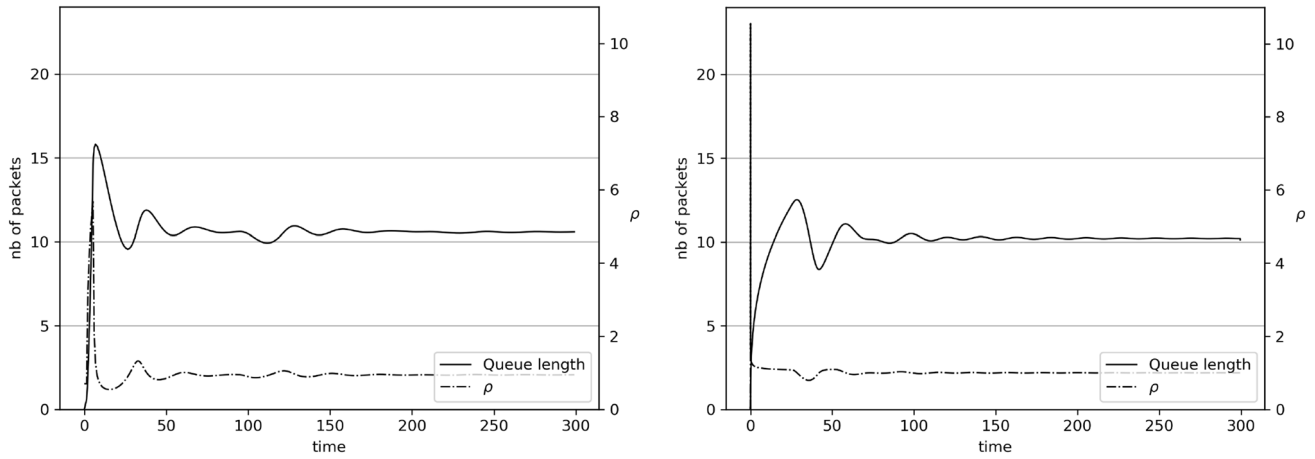


Fig. 1 Mean queue and congestion  $\rho = \lambda/\mu$ , diffusion approximation (left), fluid flow (right),  $PI^\alpha$ :  $K_p = 0.0001$ ,  $K_I = 0.0014$ ,  $\alpha = -0.8$

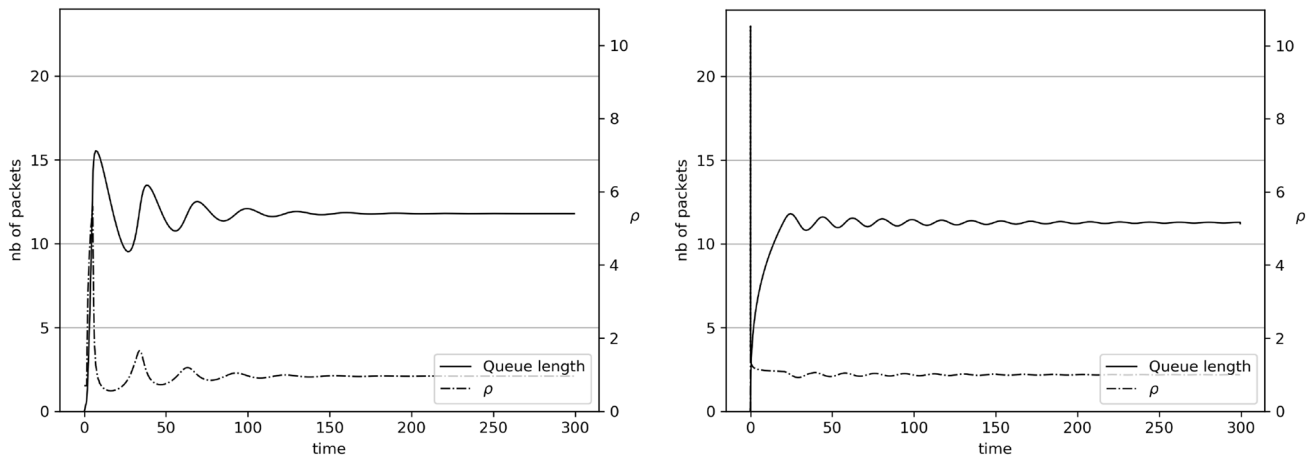


Fig. 2 Mean queue and congestion  $\rho = \lambda/\mu$ , diffusion approximation (left), fluid flow (right),  $PI^\alpha$ :  $K_p = 0.0001$ ,  $K_I = 0.0005$ ,  $\alpha = -0.4$

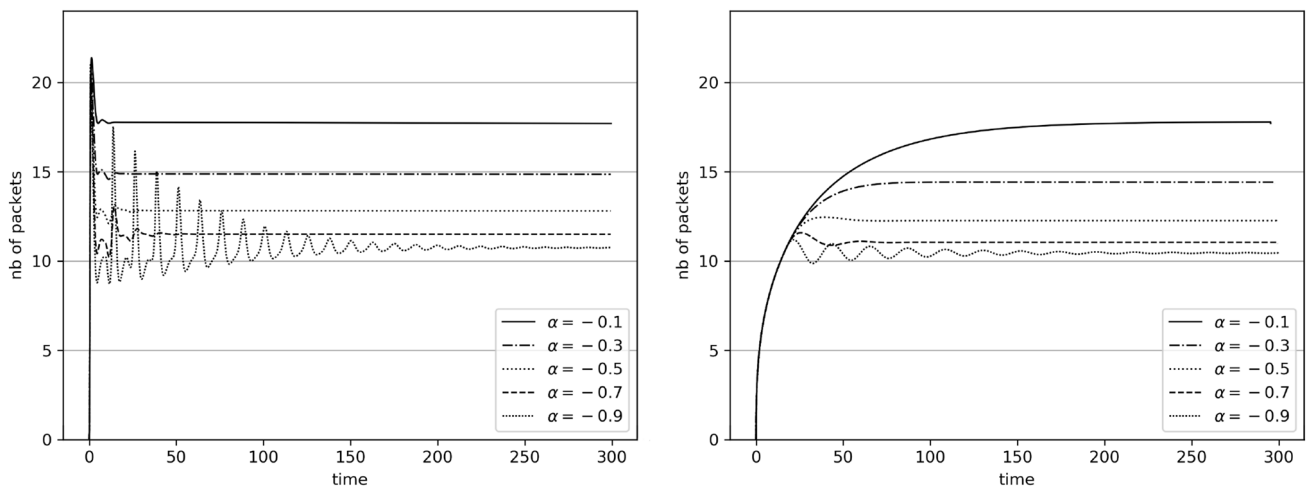


Fig. 3 Mean queue and congestion  $\rho = \lambda/\mu$ , diffusion approximation (left), fluid flow (right),  $PI^\alpha$ :  $K_p = 0.0001$ ,  $K_I = 0.0004$ ,  $\alpha$  is changing in the range  $[-0.1, -0.9]$

**Table 1** Average queue length,  $PI^\alpha: K_p = 0.0001, K_I = 0.0004$

| $\alpha$ | Fluid flow | Diffusion |
|----------|------------|-----------|
| -0.1     | 17.681     | 17.764    |
| -0.2     | 16.167     | 15.919    |
| -0.3     | 14.843     | 14.385    |
| -0.4     | 13.724     | 13.162    |
| -0.5     | 12.804     | 12.220    |
| -0.6     | 12.069     | 11.520    |
| -0.7     | 11.496     | 11.017    |
| -0.8     | 11.063     | 10.667    |
| -0.9     | 10.754     | 10.448    |

**Table 2** Average queue length, RED:  $w = 0.08$

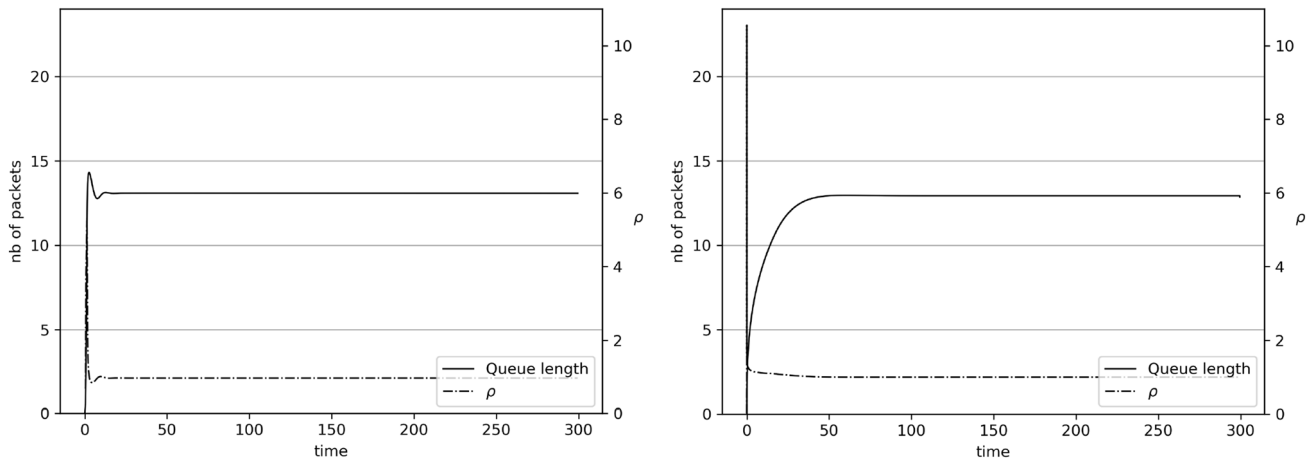
| $P_{max}$ | Fluid flow | Diffusion |
|-----------|------------|-----------|
| 0.01      | 16.884     | 16.623    |
| 0.02      | 14.595     | 14.541    |
| 0.03      | 13.548     | 13.554    |
| 0.04      | 12.922     | 12.951    |
| 0.05      | 12.497     | 12.535    |
| 0.06      | 12.186     | 12.228    |
| 0.07      | 11.948     | 11.989    |
| 0.08      | 11.759     | 11.798    |
| 0.09      | 11.605     | 11.641    |

the case of the  $PI^\alpha$  controller. In the case of RED, the mean queue oscillates between 11 and 17 packets.

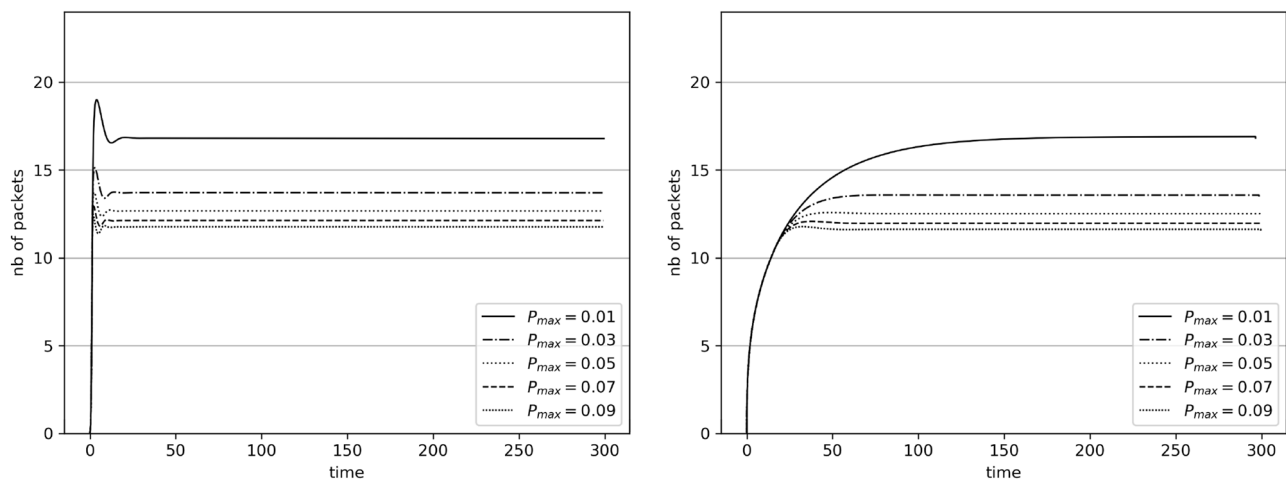
The results show that the parameters  $\alpha$  and  $P_{max}$  strongly influence the behavior of the  $PI^\alpha$  and RED controllers. The problem of the choice of parameters is still discussed, see, e.g., [13, 27]. The influence of the  $PI^\alpha$  parameters on the queue behavior was shown in [6, 9–11]. These articles describe also the advantages of  $PI^\alpha$  controllers, associated with a larger number of variable control parameters.

### Conclusions

In the article, we present two models of the TCP/AQM mechanism which are based on diffusion approximation. The AQM algorithm is either RED or  $PI^\alpha$  controller. The results of the diffusion models are compared with the output of often used fluid flow approximation models. We note a significant similarity of the results for the various RED and  $PI^\alpha$  parameters. Diffusion approximation approach gives more detailed information on the transmission delays: not only mean values, but whole distributions of these times. Therefore, we consider the proposed here model as a useful tool in TCP/IP control evaluation. In future work, we will consider more realistic Internet case studies.



**Fig. 4** Mean queue and congestion  $\rho = \lambda/\mu$ , diffusion approximation (left), fluid flow (right), RED:  $w = 0.1, p_{max} = 0.04$



**Fig. 5** Mean queue and congestion  $\rho = \lambda/\mu$ , diffusion approximation (left), fluid flow (right), RED:  $w = 0.08$ ,  $p_{max}$  is changing in the range  $[0.01, 0.09]$

**Acknowledgements** This research was partially financed by National Science Center Project no. 2017/27/B/ST6/00145.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

- Barbera M, Lombardo A, Schembra G, Trecarichi CA. Fluid flow analysis of TCP flows in a DiffServ environment. *Eur Trans Telecommun.* 2006;17:505–24. <https://doi.org/10.1002/ett.1093>.
- Braden B, Clark D, Crowcroft J, Davie B, Deering S, Estrin D, Floyd S, Jacobson V, Minshall G, Partridge C, Peterson L, Ramakrishnan K, Shenker S, Wroclawski J, Zhang L. Recommendations on queue management and congestion avoidance in the Internet. RFC 2309, IETF, 1998.
- Ciesielski M, Leszczynski J. A numerical method for solution of ordinary differential equations of fractional order. *Parallel Process Appl Math.* 2002;2328:695–702. [https://doi.org/10.1007/3-540-48086-2\\_77](https://doi.org/10.1007/3-540-48086-2_77).
- Domańska J, Augustyn DR, Domański A. The choice of optimal 3-rd order polynomial packet dropping function for NLRED in the presence of self-similar traffic. *Bull Polish Acad Sci Techn Sci.* 2012;60(4):779–86. <https://doi.org/10.2478/v10175-012-0090-x>.
- Domańska J, Domański A, Czachórski T, Klamka J. Fluid flow approximation of time-limited TCP/UDP/XCP streams. *Bull Polish Acad Sci Tech Sci.* 2014;62(2):217–25. <https://doi.org/10.2478/bpasts-2014-0021>.
- Domańska J, Domański A, Czachórski T, Klamka J. The use of a non-integer order PI controller with an Active Queue Management Mechanism. In *J Appl Math Comput Sci.* 2016;26:777–89. <https://doi.org/10.1515/amcs-2016-0055>.
- Domańska J, Domański A, Czachórski T, Klamka J. Self-similarity traffic and AQM mechanism based on non-integer order  $PI^\alpha D^\beta$  controller. *Commun Comput Inform Sci.* 2017;718:336–50. [https://doi.org/10.1007/978-3-319-59767-6\\_27](https://doi.org/10.1007/978-3-319-59767-6_27).
- Domańska J, Domański A, Czachórski T, Klamka J, Marek D, Szyguła J. GPU accelerated non-integer order  $PI^\alpha D^\beta$  controller used as AQM mechanism. *Commun Comput Inform Sci.* 2018;860:286–99. [https://doi.org/10.1007/978-3-319-92459-5\\_23](https://doi.org/10.1007/978-3-319-92459-5_23).
- Domańska J, Domański A, Czachórski T, Klamka J, Marek D, Szyguła J. The influence of the traffic self-similarity on the choice of the non-integer order  $PI^\alpha$  controller parameters. *Commun Comput Inform Sci.* 2018;935:76–83. [https://doi.org/10.1007/978-3-030-00840-6\\_9](https://doi.org/10.1007/978-3-030-00840-6_9).
- Domańska J, Domański A, Czachórski T, Klamka J, Szyguła J. The AQM dropping packet probability function based on non-integer order  $PI^\alpha D^\beta$  controller. *Lect Notes Electr Eng.* 2019;496:36–48. [https://doi.org/10.1007/978-3-319-78458-8\\_4](https://doi.org/10.1007/978-3-319-78458-8_4).
- Domańska J, Domański A, Czachórski T, Klamka J, Szyguła J, Marek D. AQM mechanism with the dropping packet function based on the answer of several  $PI^\alpha$  controllers. *Commun Comput Inform Sci.* 2019;1039:400–12. [https://doi.org/10.1007/978-3-030-21952-9\\_29](https://doi.org/10.1007/978-3-030-21952-9_29).
- Domańska J, Domański A, Czachórski Tadeusz. The drop-from-front strategy in AQM. *Next Gener Teletraffic Wired/Wirel Adv Netw* 2007;4712:61–72. [https://doi.org/10.1007/978-3-540-74833-5\\_6](https://doi.org/10.1007/978-3-540-74833-5_6).
- Feng W-C, Kandlur DD. Adaptive packet marking for maintaining end-to-end throughput in a differentiated-services internet. *IEEE/ACM Trans Netw.* 1999;7(5):685–97. <https://doi.org/10.1109/90.803383>.
- Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans Netw.* 1993;1(4):397–413. <https://doi.org/10.1109/90.251892>.
- Gelenbe E. On approximate computer systems models. *J ACM.* 1975;22(2):261–9. <https://doi.org/10.1145/321879.321888>.
- Gelenbe E, Pujolle G. The behaviour of a single-queue in a general queueing network. *Acta Inform.* 1976;7(2):123–36. <https://doi.org/10.1007/BF00265766>.
- Gelenbe E. A diffusion model for packet travel time in a random multi-hop medium. *ACM Trans Sens Netw (TOSN).* 2007;3(2):10. <https://doi.org/10.1145/1240226.1240230>.
- Gkogkas A, Rovero R, Haridi S. Accurate and efficient simulation of bandwidth dynamics for peer-to-peer overlay networks.

- In: VALUETOOLS 2011—5th International ICST Conference on Performance Evaluation Methodologies and Tools; 2011. pp. 352–361. <https://doi.org/10.4108/icst.valuetools.2011.245721>.
19. Gu Y, Liu Y, Towsley D. On integrating fluid model with packet simulation. *IEEE Infocom*. 2004;4:2856–66. <https://doi.org/10.1109/INFCOM.2004.1354702>.
  20. Haider A, Nakao A. On path switching in overlay networks. *Proc Austr Telecommun Netw Appl Confer (ATNAC)*. 2008;. <https://doi.org/10.1109/ATNAC.2008.4783350>.
  21. Heider A. Improved congestion control for packet switched data networks and the internet. *Electrical and Electronic Engineering*; University of Canterbury; 2004.
  22. Hollot CV, Misra V, Towsley D. A control theoretic analysis of RED. *IEEE/INFCOM*, pp. 1510–1519, 2001.
  23. Hollot CV, Misra V, Towsley D, Gong W. On designing improved controllers for AQM routers supporting TCP flows. In: *Twentieth Annual Joint Conference of the IEEE Computer and Communications Society. Proceedings IEEE INFOCOM*; 2001, vol 3, pp. 1726–1734. <https://doi.org/10.1109/INFCOM.2001.916670>.
  24. Krajewski W, Viaro U. On robust fractional order PI controller for TCP packet flow. *BOS Conference: Systems and Operational Research*; 2014.
  25. Kumar S, Park S, Iyengar S. A loss-event driven scalable fluid simulation method for high-speed networks. *Comput Netw Int J Comput Telecommun Netw*. 2010;54:112–32. <https://doi.org/10.1016/j.comnet.2009.08.018>.
  26. Liu C, Jain R. Improving explicit congestion notification with the mark-front strategy. *Comput Netw*. 2004;35:185–201. [https://doi.org/10.1016/S1389-1286\(00\)00167-5](https://doi.org/10.1016/S1389-1286(00)00167-5).
  27. May M, Diot C, Lyles B, Bolot J. Influence of active queue management parameters on aggregate traffic performance. *Research Report, RR-3995*, 2000.
  28. Miller KS, Ross B. *An introduction to the fractional calculus and fractional differential equations*. New York: Wiley; 1993.
  29. Misra V, Gong WB, Towsley D. Fluid-based analysis of network of AQM routers supporting TCP flows with an application to RED. In: *Computer Communication Review—CCR. SIGCOMM '00 Proceedings of the Conference on Applications, Technologies, Architectures and Protocols for Computer Communication*; 2000, vol 30(4), pp. 151–160. <https://doi.org/10.1145/347059.347421>.
  30. Newell GF. Queues with time-dependent arrival rates. I The transition through saturation. *J Appl Probab*, 1968;2(2):436–451. <https://doi.org/10.2307/3212264>.
  31. Nishioka T, Sakumoto Y, Ohsaki H, Imase M. Design and implementation of flow-level simulator for a network with heterogeneous flows. In: *2009 Ninth Annual International Symposium on Applications and the Internet*; 2009. pp. 78–84. <https://doi.org/10.1109/SAINT.2009.21>.
  32. Nycz M, Nycz T, Czachórski T. Modelling dynamics of TCP flows in very large network topologies. *Lect Notes Electr Eng*. 2016;. [https://doi.org/10.1007/978-3-319-22635-4\\_23](https://doi.org/10.1007/978-3-319-22635-4_23).
  33. Padhy PK, Sundaram RK. Analysis and design of improved PI-PD controller for TCP AQM routers. *Int Confer Power, Control Embedded Syst*. 2010. <https://doi.org/10.1109/ICPCE S.2010.5698714>.
  34. Podlubny I. *Fractional differential equations*. San Diego: Academic Press; 1999.
  35. Podlubny I. Fractional order systems and  $PI^\lambda D^\mu$  controllers. *IEEE Trans Autom Control*. 1999;44(1):208–14.
  36. Sakumoto Y, Ohsaki H, Imase M. On XCP stability in a heterogeneous network. In: *12th IEEE Symposium on Computers and Communications*; 2007. pp. 531–537. <https://doi.org/10.1109/ISCC.2007.4381611>.
  37. Shah PA, Yasmin S, Asghar S, Qayyum A, Hasbullah HB. A fluid flow model for SCTP traffic over the internet. In: *Proceedings of the International Conference on Emerging Technologies (ICET)*; 2012. pp. 1–6. <https://doi.org/10.1109/ICET.2012.6375481>.
  38. Czachórski T. A method to solve diffusion equation with instantaneous return processes acting as boundary conditions. *Bull Polish Acad Sci Techn Sci*. 1993;41(4):417–51.
  39. Towsley D, Gong W, Hollot K, Liu Y, Vishal M. Fluid methods for modeling large: heterogeneous networks. *NTIS*; 2005.
  40. Li W, Zeng-zhi L, Yan-ping C, Ke X. Fluid-based stability analysis of mixed TCP and UDP traffic under RED. In: *10th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'05)*; 2005. pp. 341–348. <https://doi.org/10.1109/ICECC S.2005.45>.
  41. Zheng B, Atiquzzaman M. DSRED: a new queue management scheme for the next generation internet. *IEICE Trans Commun*. 2000;. <https://doi.org/10.1093/ietcom/e89-b.3.764>.







**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## **Rozdział 9**

# **Diffusion Model of a Non-Integer Order $PI_\gamma$ Controller with TCP/UDP Streams**

Article

# Diffusion Model of a Non-Integer Order $PI^\gamma$ Controller with TCP/UDP Streams

Dariusz Marek <sup>1</sup>, Adam Domański <sup>1</sup>, Joanna Domańska <sup>2</sup>, Jakub Szygula <sup>1,\*</sup>, Tadeusz Czachórski <sup>2</sup>  
and Jerzy Klamka <sup>2</sup>

<sup>1</sup> Department of Distributed Systems and Informatic Devices, Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland; dariusz.marek@polsl.pl (D.M.); adam.domanski@polsl.pl (A.D.)

<sup>2</sup> Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland; joanna@iitis.pl (J.D.); tadek@iitis.pl (T.C.); jerzy.klamka@iitis.pl (J.K.)

\* Correspondence: jakub.szygula@polsl.pl

**Abstract:** In this article, a way to employ the diffusion approximation to model interplay between TCP and UDP flows is presented. In order to control traffic congestion, an environment of IP routers applying AQM (Active Queue Management) algorithms has been introduced. Furthermore, the impact of the fractional controller  $PI^\gamma$  and its parameters on the transport protocols is investigated. The controller has been elaborated in accordance with the control theory. The TCP and UDP flows are transmitted simultaneously and are mutually independent. Only the TCP is controlled by the AQM algorithm. Our diffusion model allows a single TCP or UDP flow to start or end at any time, which distinguishes it from those previously described in the literature.

**Keywords:** active queue management; diffusion approximation; fractional controller  $PI^\gamma$ ; internet; TCP/IP and UDP



**Citation:** Marek, D.; Domański, A.; Domańska, J.; Szygula, J.; Czachórski, T.; Klamka, J. Diffusion Model of a Non-Integer Order  $PI^\gamma$  Controller with TCP/UDP Streams. *Entropy* **2021**, *23*, 619. <https://doi.org/10.3390/e23050619>

Academic Editor: José A. Tenreiro Machado

Received: 24 April 2021

Accepted: 13 May 2021

Published: 16 May 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Network protocols control the transfer of information between a transmitter and receiver. They have to fulfil many functionalities, for example, ensure the correctness of transmitted data and increase channel capacity. This article focuses on the problem of transmission efficiency. Traffic intensity has a stochastic nature. The protocols should adapt the transmission to the limited bandwidth and deterministic mechanisms of computer networks. The goal is to minimize information loss due to packet drops in overflowing buffers. These control activities are based on the exchange of information between sender and receiver. The article analyses two approaches to increase the efficiency (by minimizing packet loss) in the wide-area network:

- a mechanism for controlling the speed of sending information at the transmitter level, due to the TCP protocol [1];
- removing data from the network regulated by the AQM mechanism on the IP level [2].

Due to the rapid development of network communication technology, more and more attention has been focused on the problems of congestion control. Congestion is an important factor affecting a network's Quality of Service (QoS) and reducing its performance. The Transport Control Protocol (TCP) has already been used in Internet applications for more than 30 years. During this time, several congestion control algorithms have been developed to meet the requirements of a constantly changing computer network. Active Queue Management (AQM) is a network approach to congestion prevention that works in combination with the TCP protocol. The most effective congestion control occurs when the AQM mechanism and TCP protocol work together [3].

The earliest of the AQM algorithms is called Random Early Detection (RED) [4]. In the past twenty years, many AQM mechanisms have been proposed. These mechanisms

can be classified into three categories [5,6]: heuristic, optimization and control theory approach. The heuristic approach towards the AQM mechanism heavily depends on intuition. Examples of such mechanisms are inter alia, BLUE algorithm [7], hyperbola RED (HRED) [8], and Yellow algorithm [9]. These algorithms aim to improve packet loss, fairness, network utilization, and adaptability to different characteristics of the network traffic. The AQM based on the optimization approach was developed by Frank Kelly. His paper [10] transforms the design of the AQM algorithm to a convex optimization problem. Another AQM algorithm elaborated following the optimization theory was the Random Early Marking (REM) [11]. However, the AQM mechanisms based on the optimization theory are usually complex and hard to tune [6]. In this case, it is also difficult to control the instantaneous queue length of the router [12]. To overcome these effects, researchers have resorted to the control theory. The exact type of a  $PI^\gamma$  controller used in this article belongs to the family of AQM mechanisms based on this theory, described in Section 2.

The dynamic model of the TCP behavior is required to enable the application of the control theory principles to AQM. The transient analysis should be performed in the model to provide us with time-dependent behavior of flows and queues. Apart from discrete time simulation, which is very time-consuming in the case of transient-state analysis, fluid flow or diffusion approximation methods can be used. Usually, due to its simplicity, the fluid flow method is the one used to model TCP networks [13–16]. The diffusion approximation, on the other hand, offers more accurate results. In the diffusion approximation model, traffic flows are determined by their mean and variance. On top of that, second-order partial equations describe queue changes.

In this paper, the transient behavior of the AQM mechanisms is analyzed via extending our earlier models presented in [17]. Comparing with previous results, the consideration of independent TCP/UDP streams that can start and end at any time constitutes a novelty. The diffusion approximation method is employed to trace the behavior of the AQM mechanisms applied to control Internet traffic. We are able to describe the features of RED, NLRED, PI and  $PI^\gamma$  that were not possible to notice in an open-loop or a fluid-flow scenario. Table 1 presents main approaches to AQM based on PI and non-integer order  $PI^\gamma$  controllers.

**Table 1.** Main approaches of AQM based on PI and non-integer order  $PI^\gamma$  controllers.

|  |  |
|--|--|
| PI (Simulation) [18]                                     | Study of the TCP/AQM mechanisms based on PI controllers  |
| PID (Simulation) [19]                                    | Evaluation of the AQM based on non-integer order PID controller  |
| $PI^\gamma$ (Fluid-Flow) [20]                            | First application of non-integer order $PI^\gamma$ controller to an AQM strategy   |
| $PI^\gamma$ (Fluid-Flow/Simulation) [21]                 | Fluid flow approximation and discrete-event simulation to investigate the influence of the AQM policy based on non-integer order $PI^\gamma$ controller on the packet loss probability, the queue length and its variability                         |
| $PI^\gamma D^\omega$ (Simulation) [22]                   | Model of AQM mechanism based on non-integer order $PI^\gamma D^\omega$ controller  |
| $PI^\gamma$ (Simulation) [23]                            | Finding optimal parameters of the non-integer order $PI^\gamma$ controller used as AQM mechanisms. The optimization was made by using the well-known Hooke and Jeeves direct search method applied for minimization of a multivariate score function |
| Adapted $PI^\gamma$ (Simulation) [24]                    | Choice of non-integer order $PI^\gamma$ controller parameters based on machine learning algorithms. The controller parameters automatically adjust to network traffic parameters (traffic intensity and self-similarity)                             |
| TCP $PI^\gamma$ (Diffusion) [17]                         | The diffusion approximation model of the simple TCP traffic. Evaluation (in close loop scenario) of the effectiveness of active queue management (AQM) mechanisms based non-integer order $PI^\gamma$ controller                                     |
| TCP $PI^\gamma$ (Combined Diffusion and Simulation) [25] | Combined diffusion approximation and simulation model based on non-integer order $PI^\gamma$ controller  |

This paper is organized as follows: In Section 2 one can find the literature overview of AQM mechanisms based on the control theory approach. This section also describes an embedding of our model in the area of diffusion modelling. Section 3 gives a brief description of the AQM mechanism used in this paper. Section 4 describes our diffusion

model of the  $PI^\gamma$  controller with TCP/UDP streams. In Section 5, numerical results are presented. Section 6 concludes our work.

## 2. Background and Related Work

### 2.1. AQMs Based on the Control Theory Approach

Yes, it should be subsection, thank you. The detection and mitigation of congestion are some of the biggest problems in the computer networks domain [3]. Many mechanisms have been created to solve these issues. They are based on different operating principles. Most of the congestion avoidance mechanisms have been implemented in transport layer protocols such as TCP [26]. UDP applications, such as voice or video traffic, do not have any congestion avoidance mechanisms. In the case of such applications, it is possible to use the open-loop hop-by-hop backpressure strategy. In accordance with this strategy, a congested node broadcasts backpressure messages to upstream nodes to reduce their transmission rate [27]. This method is useful for bandwidth control but, unfortunately, introduces transmission delays. Therefore, there are many attempts to improve these kinds of algorithms, for example, [28]. Back pressure algorithms can be used in wireless sensor networks (WSNs) because of their limited need for: computation resources, storage, energy and communication bandwidth [27,29]. The TCP protocol and the back pressure mechanism are incompatible due to a mismatch between the TCP congestion control mechanism and the back pressure queue size based routing [26]. As a result, wider use of this solution in the Internet would require changes in the TCP protocol [26].

The traffic control in TCP/IP network is, in fact, a closed-loop algorithm in which AQM algorithm plays the role of a controller. The performance and dynamics of network connections can be studied using the control theory to improve their stability and to reduce the reaction time. Several feedback control algorithms have been developed. The article [30] proposes a fluid flow dynamic model of TCP/RED networks by using stochastic differential equations. Based on this dynamic model, several AQM controllers have been proposed using different control approaches. The article [13] proposes a Proportional-Integral (PI) controller on low-frequency dynamics. Authors of the article [31] propose adaptive Proportional (P) and Proportional-Integral (PI) controller and conclude that PI controller can adapt very well to the large fluctuation of the Internet traffic. The article [32] describes a new variant of the RED mechanism called Proportional-Derivative-RED (PD-RED) that performs better than Adaptive RED. Authors of [19] propose the Proportional-Integral-Differential (PID) controller to speed up the responsiveness of the AQM system. Among them, PI controllers are attracting increased attention because of their computation, and implementation simplicity [18]. The article [6] tries to preserve the simplicity of the PI controller by proposing a self-tuning compensated PID controller.

Traditional calculus is based on integer-order differentiation and integration. Differentiation or integration of non-integer order have been used in many mathematical models of dynamic systems. The article [33] claims that many real dynamic systems are better characterized using a fractional dynamic model. Authors indicate that non-integer order controllers provide better performance than the conventional integer order ones. The article [20] presents the first application of the fractional order PI controller to an AQM strategy. The authors focus on the method for determining the parameter regions where the  $PI^\gamma$  controller ensures a given modulus margin (inverse of the  $H_\infty$  norm of the sensitivity function). The article [21] describes an evaluation of the fractional-order  $PI^\gamma$  controller used as an AQM mechanism. The performance of the controller is evaluated using fluid flow approximation (closed-loop control) and simulation (open loop scenario). The article [34] studies the proper selection of the  $PI^\alpha D^\beta$  parameters to show an influence of the proportional, integral and derivative terms on the controller's dropping function. A simulation model is used in this article. The article [17] proposes a new model of the  $PI^\gamma$  controller based on a diffusion approximation approach. This model is able to provide more detailed information on transmission delays than the frequently used fluid flow model.



### 2.2. Diffusion Approximation

Diffusion models refer to the changes of flows defined by their mean and variance. They are more accurate than the fluid flow models, where only the mean value is considered. They use the central limit theorem to justify that the number of arrivals and services at a queueing system tends to the normal distribution and a diffusion process may represent the queue length. The solution of the diffusion equation with parameters depending on the analyzed system approximates a queue distribution. This way, the models easily incorporate general distributions of interarrival and service times and the transient queue behavior.

The diffusion approximation has been used to study the performance evaluation of computer systems and networks for many years. The tutorial [35] describes how the diffusion approximation formalism can be applied to the analysis of some traffic control mechanisms in the ATM network. The article [36] presents the diffusion model of wireless network based on the IEEE 802.11 protocol. In the article [37], the first attempt to model the TCP/RED router using diffusion approximation is made. The article [38] applies diffusion approximation to model the influence of a buffer capacity on Quality of Experience in wireless video connections. In our article [17], a new model of TCP NewReno based on the diffusion approximation method is developed. The combined diffusion approximation and simulation model is proposed in [25]. These models allow analyzing the behavior of a single TCP stream. To the best of our knowledge, there is no diffusion model described in the literature which can be used to analyze the independent TCP/UDP streams.

### 3. RED, NLRED and a Non-Integer Order $PI^\gamma$ Controllers

The traffic control built-in a TCP/IP protocol is a typical closed-loop one. The flow of packets emitted by a sender is controlled by the loss probability  $p$  observed in routers and reported to the sender with a certain delay. The losses decrease the traffic, and their lack increases it. In the case of a passive router, the losses occur when the router queue is full.

The RED family algorithms determine the dropping probability even if there is still a place to store packets but the queue increases. They use the weighted moving average  $avg$  computed at packets' arrival; for packet  $i$ :

$$avg_i = (1 - w)avg_{i-1} + wq_i$$

where  $avg_{i-1}$  is the moving average computed at the arrival of the previous packet, and  $q_i$  is the queue seen by the packet  $i$ . The probability of a loss is for small values of  $avg$   $p_{RED} = 0$ . As  $avg$  grows, the probability increases linearly between two thresholds  $Min_{th}$  and  $Max_{th}$ , from 0 to  $p_{max}$ .

$$P_{RED}(avg) = p_{max} \frac{avg - Min_{th}}{Max_{th} - Min_{th}} \tag{1}$$

Finally, it becomes  $p_{RED} = 1$  given  $avg > Max_{th}$ .

Another modification of RED is NLRED (non-linear RED) algorithm [39]. In this mechanism, the linear packet dropping function is replaced by a quadratic function:

$$p = \begin{cases} 0 & \text{for } avg < Min_{th} \\ (\frac{avg - Min_{th}}{Max_{th} - Min_{th}})^2 P_{max} & \text{for } Min_{th} \leq avg \leq Max_{th} \\ 1 & \text{for } avg > Max_{th} \end{cases} \tag{2}$$

Paper [40] proposes another approach to a non-linear packet dropping function. This function is based on the third-degree polynomials instead of the well-known quadratic function. This approach allows to choose the optimal packet dropping function:

$$p(avg, a_1, a_2, p_{max}) = \begin{cases} 0 & \text{for } avg < Min_{th} \\ \varphi_0(avg) + a_1 \varphi_1(avg) + a_2 \varphi_2(avg) & \text{for } Min_{th} \leq avg \leq Max_{th} \\ 1 & \text{for } avg > Max_{th} \end{cases} \tag{3}$$

where:

$$\varphi_0(avg) = p_{max} \frac{avg - Min_{th}}{Max_{th} - Min_{th}}, \tag{4}$$

$$\varphi_1(avg) = (avg - Min_{th})(Max_{th} - avg), \tag{5}$$

$$\varphi_2(avg) = (avg - Min_{th})^2(Max_{th} - avg) \tag{6}$$

Probability  $p$  can also be determined directly by the controller by comparing the current queue  $q_i$  and the queue  $q_0$  we wish to maintain. Their difference, called error in the control theory, is the input signal to the controller, which may be the classical PID one or, as it has been investigated recently, the proportional-integral  $PI^\gamma$  of fractional, that is, non-integer order. Fractional derivatives and integrals are known since the times of Leibnitz and recently become used in the control of physical processes [41].

In the case of the  $PI^\gamma$  controller, the loss probability  $p_i$  of a packet  $i$  is equal to:

$$p_i = \max\{0, -(K_p e_i + K_I \Delta^\gamma e_i)\} \tag{7}$$

It depends on the proportional and integral terms  $K_p, K_I$ , the error  $e_i = q_i - q$ , and the order of integration  $\gamma$ . Their impact is further discussed in the articles [21,22,34,42].

The packet drop probability is determined at discrete moments of packet arrivals. There exists only one definition of the non-integer order discrete differ-integral. It [43] is a generalization of the traditional definition of the difference between the integer-order and the non-integer one and is analogous with the generalization employed in the Grünwald-Letnikov (GrLET) formula [44,45].

For a given sequence  $f_0, f_1, \dots, f_j, \dots, f_k$

$$\Delta^\gamma f_k = \sum_{j=0}^k (-1)^j \binom{\gamma}{j} f_{k-j} \tag{8}$$

where  $\gamma \in R$  is by and large a non-integer fractional order,  $f_k$  is a differentiated discrete function, and  $\binom{\gamma}{j}$  is a generalised Newton symbol which definition looks as follows:

$$\binom{\gamma}{j} = \begin{cases} 1 & \text{for } j = 0 \\ \frac{\gamma(\gamma - 1)(\gamma - 2) \dots (\gamma - j + 1)}{j!} & \text{for } j = 1, 2, \dots \end{cases} \tag{9}$$

Parameters of the non-integer order  $PI^\gamma$  controller are presented in Table 2.

**Table 2.** Parameters of the non-integer order  $PI^\gamma$  controller.

|          |                       |
|----------|-----------------------|
| $K_p$    | Proportional term     |
| $K_I$    | Integral term         |
| $\gamma$ | Integral order        |
| $e_i$    | Error in current slot |
| $q_i$    | Actual queue length   |
| $q$      | Desired queue length  |

#### 4. Diffusion Approximation of the TCP and UDP Network Streams

In this section, how to model the AQM router supporting TCP/UDP flows using the diffusion approximation is described. The main goal of the analysis presented below is to model the Active Queue Management based on the answer of  $PI^\gamma$  controller.

The method of diffusion approximation is used in queueing theory (e.g., [46–49]) when it is hard to determine a queue distribution. The queue length is replaced by the value of diffusion process  $X(t)$ . The probability density function (pdf) of the letter,  $f(x, t; x_0)$

$$f(x, t; x_0) dx = P[x \leq X(t) < x + dx | X(0) = x_0] \tag{10}$$

of  $X(t)$ , is given by the diffusion equation:

$$\frac{\partial f(x, t; x_0)}{\partial t} = \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x} \tag{11}$$

and helps us to evaluate the queue distribution. It is usually used in the analysis of G/G/1 queueing systems, that is, having general distributions of interarrival and service times or G/G/1/L, where additionally the queue is limited to L positions. In the latter case the number of packets in router is in the range  $[0, L]$  and therefore the diffusion process is limited by barriers in  $x = 0$  and  $x = L$ , and the diffusion equation is used in the form [47]:

$$\begin{aligned} \frac{\partial f(x, t; x_0)}{\partial t} = & \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x} \\ & + \lambda p_0(t) \delta(x - 1) + \lambda p_L(t) \delta(x - L + 1) \end{aligned}$$

$$\begin{aligned} \frac{dp_0(t)}{dt} = & \lim_{x \rightarrow 0} \left[ \frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0) \right] - \lambda p_0(t), \\ \frac{dp_L(t)}{dt} = & \lim_{x \rightarrow L} \left[ \frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0) \right] - \mu p_L(t). \end{aligned} \tag{12}$$

In the above equations,  $p_0(t)$ ,  $p_L(t)$  denote probabilities that the process is in either of barriers,  $\lambda$  and  $\mu$  are the intensities of jumps from barriers; from  $x = 0$  to  $x = 1$  and from  $x = L$  to  $x = L - 1$  corresponding to the arrival of a packet to the empty queue or departure of a packet from the full queue. Parameters  $\beta$  and  $\alpha$  are chosen as  $\beta = \lambda - \mu$ ,  $\alpha = \lambda^3 \sigma_A^2 + \mu^3 \sigma_B^2$ , where  $1/\lambda$ ,  $1/\mu$  are the first moments of interarrival and service time distributions, and  $\sigma_A^2$ ,  $\sigma_B^2$  are their variances, more information please see Table 3. This way the changes of the diffusion process and of the queue have the same mean and variance. The steady-state solution of Equation (12) is given in [47] and the transient case is considered in [49].

**Table 3.** Main notations and parameters of the diffusion model.

|                |  |
|----------------|--|
| $\lambda$      | Intensity of the input traffic   |
| $\mu$          | Intensity of packet processing and dispatching   |
| $\sigma_A^2$   | Variance of interarrival time distribution   |
| $\sigma_B^2$   | Variance of service time distribution  |
| $C_A^2$        | Squared coefficient of variation of interarrival time distribution                                   |
| $C_B^2$        | Squared coefficient of variation of service time distribution  |
| $X(t)$         | Diffusion process  |
| $\beta$        | Diffusion parameter; $\beta dt$ is the mean value of changes of $X(t)$ during $dt$                   |
| $\alpha$       | Diffusion parameter; $\alpha dt$ is the variance of changes of $X(t)$ during $dt$                    |
| $f(x, t, x_0)$ | Probability density that the process will be in state $x$ at time $t$ , for initial conditions $x_0$ |

Below, we expand this model to include a number of independent streams. We assume a queue supports  $k$  input streams. For each stream, packets arrive at intervals which are described by the distribution  $A^{(k)}(x)$ . The service time distribution is equal to  $B^{(k)}(x)$ . The  $k$  input stream ( $k = 1, \dots, K$ ) is described by the distribution  $A^{(k)}(x)$ , (with the average value  $1/\lambda^{(k)}$  and variance  $\sigma_A^{(k)2}$ ). The service time of the  $k$  stream has a distribution  $B^{(k)}(x)$  with the average value  $1/\mu^{(k)}$  and variance  $\sigma_B^{(k)2}$ . The density functions of these distributions are denoted by  $a^{(k)}(x)$  i  $b^{(k)}(x)$ . We assume input streams of individual

classes are independent. The normal distribution of the number of packets of a  $k$ -th stream coming over a period of time is approximately equal to  $\lambda^{(k)}t$ , with a variance satisfying the equation  $\lambda^{(k)3}\sigma_A^{(k)2}t = \lambda^{(k)}C_A^{(k)2}t$ . The number of packets of all streams that arrived during this time has also a normal distribution with the average value  $\lambda t = \sum_{k=1}^K \lambda^{(k)}t$  and a variance satisfying the equation  $\lambda C_A^2 t = \sum_{k=1}^K \lambda^{(k)}C_A^{(k)2}t$ . So, the parameters of the totality of streams are:

$$\lambda(t) = \sum_{k=1}^K \lambda^{(k)}(t), \quad C_A^2 = \sum_{k=1}^K \frac{\lambda^{(k)}(t)}{\lambda(t)} C_A^{(k)2} \tag{13}$$

where  $\frac{\lambda^{(k)}(t)}{\lambda(t)}$  is the probability that a given packet belongs to a stream  $k$  which allows us to determine the resultant service time parameters:

$$\frac{1}{\mu} = \sum_{k=1}^K \frac{\lambda^{(k)}}{\lambda} \frac{1}{\mu^{(k)}}, \quad C_B^2 = \mu^2 \sum_{k=1}^K \left[ \frac{\lambda^{(k)}}{\lambda} \frac{1}{\mu^{(k)2}} (C_B^{(k)2} + 1) \right] - 1, \tag{14}$$

and then the parameters  $\alpha, \beta$  of the diffusion equations:

$$\beta(t) = \lambda(t) + \mu, \quad \alpha(t) = \lambda(t)C_A^2 + \mu C_B^2$$

The distribution  $p(n) \approx f(n)$  specifies the number of packets of all streams in the queue, and the probability that there are  $v$  packets which belong to the  $k$ -th stream in the queue equals to:

$$p^{(k)}(v) = \sum_{n=v}^{\infty} [p(n) \binom{n}{v} \left(\frac{\lambda^{(k)}}{\lambda}\right)^v (1 - \frac{\lambda^{(k)}}{\lambda})^{n-v}], \tag{15}$$

where  $k = 1, \dots, K$ .

In the article, the case of two kinds of input streams is presented. UDP stream is a CBR stream with assumed number of packets sent per time unit ( $\lambda_{UDP}(t)$  is constant) and TCP stream for which the input intensity changes according to the TCP NewReno congestion control algorithm ( $\lambda_{TCP}(t)$  is shaped by the AQM mechanism). When the diffusion model is considered, the total intensity of the input stream is equal to the sum of intensities of the components.

In the case of two input streams, one TCP and one UDP, the mean and the variance of input distribution for a queue is calculated as follows:

$$\begin{aligned} \beta(t) &= \lambda_{TCP}(t) + \lambda_{UDP}(t) + \mu \\ \alpha(t) &= [\lambda_{TCP}(t) + \lambda_{UDP}(t)] \\ &\quad \left[ \frac{\sigma_{A_{TCP}}^2(t)\lambda_{TCP}^3(t) + \sigma_{A_{UDP}}^2(t)\lambda_{UDP}^3(t)}{\lambda_{TCP}(t) + \lambda_{UDP}(t)} \right] + \mu C_B^2 \\ \alpha(t) &= \sigma_{A_{TCP}}^2(t)\lambda_{TCP}^3(t) + \sigma_{A_{UDP}}^2(t)\lambda_{UDP}^3(t) + \mu C_B^2 \end{aligned} \tag{16}$$

The TCP NewReno/AQM model based on the diffusion approximation works as follows: The diffusion approximation gives the distribution of the router's queue at time  $t$ . The mean value of this queue length modifies the packet rejection probability. This probability affects the intensity  $\lambda$  of the input stream because the congestion window increases by one for lossless transmission or halves for packet loss. We assume that the flow  $\lambda$ , the size of the congestion window  $W$ , and the delay  $q/\mu$  of the router are linked together in the following way:

$$\lambda = \frac{W\mu}{q} \tag{17}$$

where  $1/\mu$  is the mean service time. In the diffusion model, the calculations are carried out inside the intervals of length  $\Delta t = 1/\lambda$ . When  $\lambda$  changes, the length of the interval changes as well. We obtain a queue distribution at time  $t_i$  at the end of the interval  $i$ . Based on it, we get the average queue length  $E[q_i]$  and then the probability of packet rejection  $p_i$  which defines new value of  $\lambda$ :  $\lambda_{i+1} = \lambda_i + \Delta\lambda_i$  due to the AQM algorithm, where

$$\Delta\lambda_i = \frac{\mu_i}{E[q_i]} - \frac{\lambda_i^2}{2} \frac{E[q_i]}{\mu} p_i. \tag{18}$$

Over the time  $t_{i+1} = t_i + 1/\lambda_i$ , we repeat the calculations for a new value of  $\lambda$ . In the model presented in this article it is assumed that the  $i$ -th flow can start or end a transmission at any moment. The change in the source intensity  $\Delta\lambda$  in the time  $t$  affects the dispatching time and the queue length. The algorithm of these calculations is presented in Figure 1.

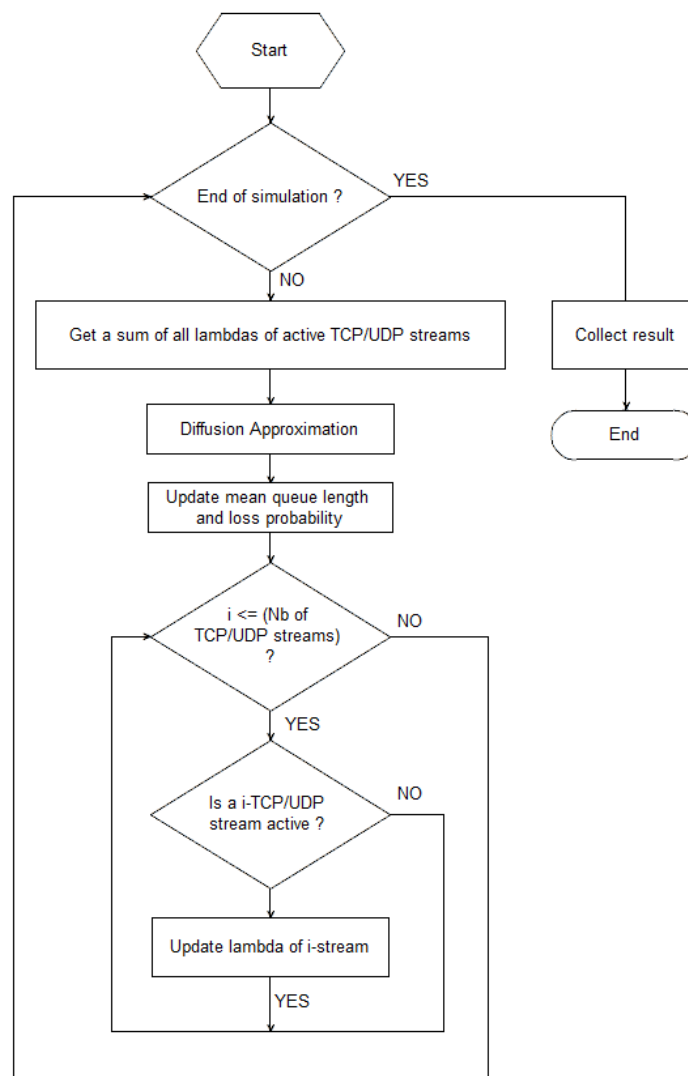


Figure 1. Flowchart of our model behavior.

### 5. Numerical Results

In this chapter, the results obtained through the method described above are discussed. We employ the proposed numerical scheme to obtain the transient behavior of TCP window

dynamics and queue length. The results for various types of input streams and AQM mechanisms (RED, NLRED, PI,  $PI^\gamma$ ) are discussed.

We assume the following parameters of the AQM buffer:  $Min_{th} = 10$ ,  $Max_{th} = 20$ , buffer size (measured in packets) = 30.

The parameters of RED are as follows: weight parameter  $w = 0.008$ ,  $p_{max} = 0.02$ . Article [50] shows the impact of RED parameters on the network traffic. Using higher values of these parameters results in the increase in network traffic fluctuations. This choice of RED parameters shows more clearly the influence of TCP/UDP flows on the AQM queue behavior [17]. These parameters are slightly different than those proposed in the literature [51]. The NLRED parameters are as follows:  $a_1 = 0.00042$ ,  $a_2 = -0.0000038$  and  $p_{max} = 0.855$ . Article [40] proposes using parameters with such values to achieve the best transmission performance.

The parameters of  $PI^\gamma$ :  $setpoint = 10$ ,  $P = 0.0001$ ,  $I = 0.005$ ,  $\gamma = -0.4$ . For  $\gamma = -1.0$  the PI controller is considered (Table 4). The proper choice of AQM/PI controller parameters is difficult. It strongly affects the packet dropping function (i.e., integral order  $\gamma$  accelerates and strengthens the controller's response). Proper selection of AQM parameters should help achieve two goals: obtaining desired queue behavior and adaptation to network transmission conditions. The influence of the controller's parameters on queue behavior is discussed in papers [21,22].

Figures 2–17 show the evolution of sources intensity and queue length. In Figure 2 the queue behavior in the case of one TCP flow and  $PI^\gamma$  controller is shown. The source intensity increases until the queue reaches the desired length. Exceeding the desired queue length increases the probability of a packet loss in AQM and thus causes reduction in the source intensity. Consequently, the queue occupancy and packet loss probability decreases. So, the source intensity increases after some time. Source intensity and queue length oscillations continue until a stable state is reached. In the stable state, the average queue length reaches 10.2 packets, and the source intensity reaches  $\lambda = 1$ . In the case of a PI controller (Figure 2) the average queue length reaches almost the same value (10.01 packets). Nevertheless, the queue occupancy still oscillates between extreme values. Such oscillations, associated with a big fractional order of the controller, are described in the article [24].

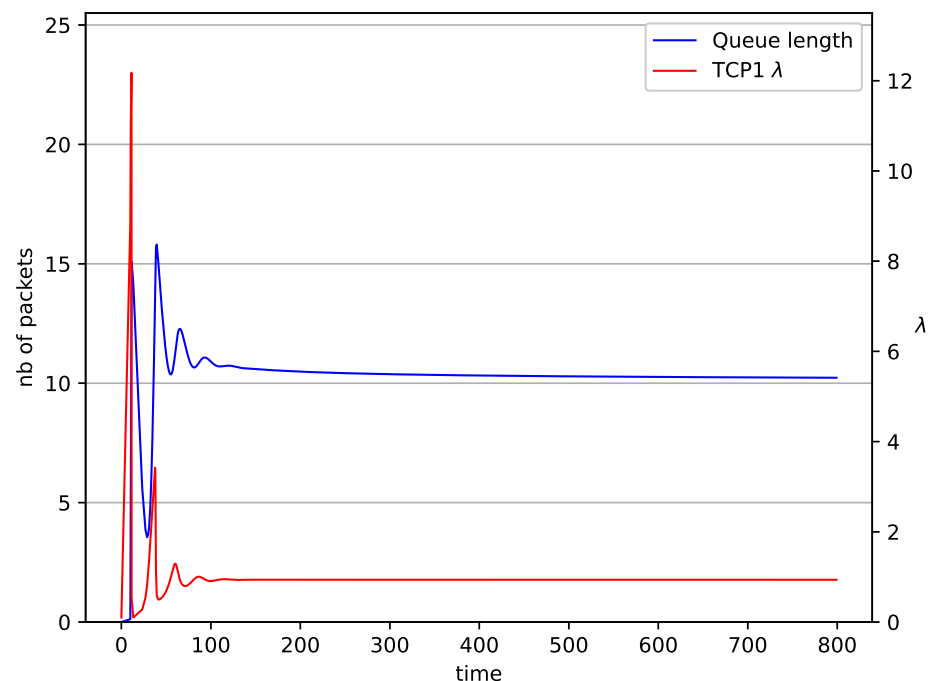


Figure 2. 1 TCP with  $PI^\gamma$  controller.

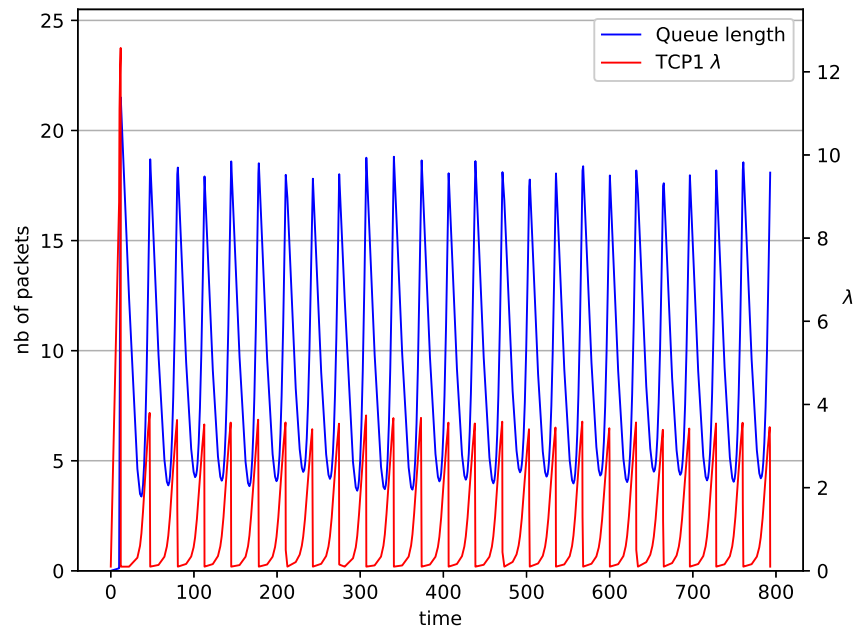


Figure 3. 1 TCP with PI controller.

Table 4.  $PI^\gamma$  and PI controllers coefficients.

|   | $K_p$  | $K_i$ | $\gamma$ | Setpoint | Type of Controller           |
|---|--------|-------|----------|----------|------------------------------|
| 1 | 0.0001 | 0.005 | -0.4     | 10       | non integer order controller |
| 2 | 0.0001 | 0.005 | -1.0     | 10       | classical controller         |

The behaviors of a single TCP stream in cooperation with the RED algorithm (Figure 4) and NLRED (Figure 5) are shown. It is identical to the previous one (Figure 2). The average queue length reaches 14.7 for RED and 10.5 packets for NLRED, the source intensity reaches  $\lambda = 1$ . This value is roughly halfway between  $Min_{th}$  and  $Max_{th}$ .

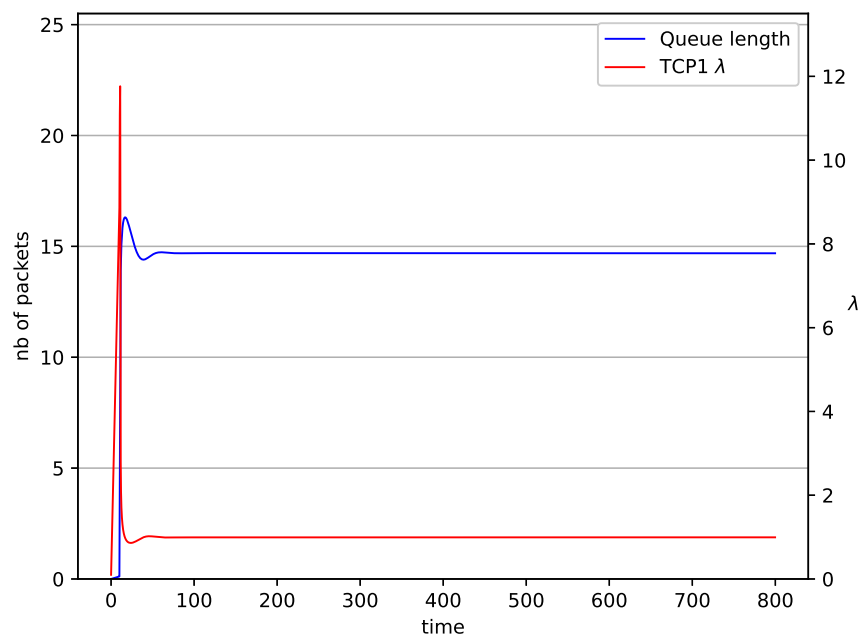


Figure 4. 1 TCP with RED controller.

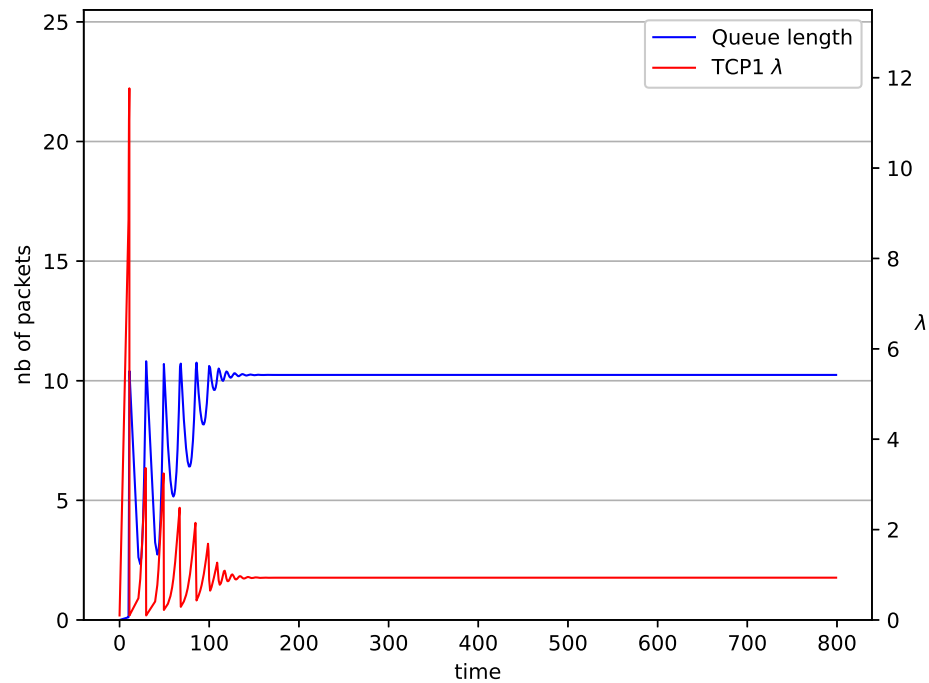


Figure 5. 1 TCP with NLRED controller.

In Figures 6–17 the behavior of the TCP/UDP AQM system for several co-transmitted data streams is portrayed.

In Figures 6–9 simultaneous transmission of TCP and UDP streams is presented. Figure 6 shows the situation where a queue is controlled by the  $PI^\gamma$  mechanism. The TCP transmitter starts a transmission at time  $t = 0$ . The TCP window evolution proceeds, as shown in Figure 2. At  $t = 200$  the UDP stream (constant intensity  $\lambda = 0.5$ ) starts its transmission. Thereby, the intensity of the TCP stream is reduced and the queue occupancy is increased. Over time, the controller attempts to stabilize the queue at the desired level. At  $t = 650$ , the UDP stream ends its transmission. As a result, the queue length decreases. The end of the UDP transmission also causes the TCP window to grow. Consequently, the queue length increases up to the desired size. When it comes to the  $PI$  controller (Figure 7), queue fluctuations occur as previously. They disappear when an additional UDP stream is loaded into the system and reappear once it has finished the transmission. This behavior is also visible in the next experiments. A similar layout of input streams is presented in Figures 8 and 9. In the case of the RED and NLRED algorithms, the commencement of the UDP stream increases the queue length. For the RED mechanism, it reaches the value near  $Max_{th}$ .



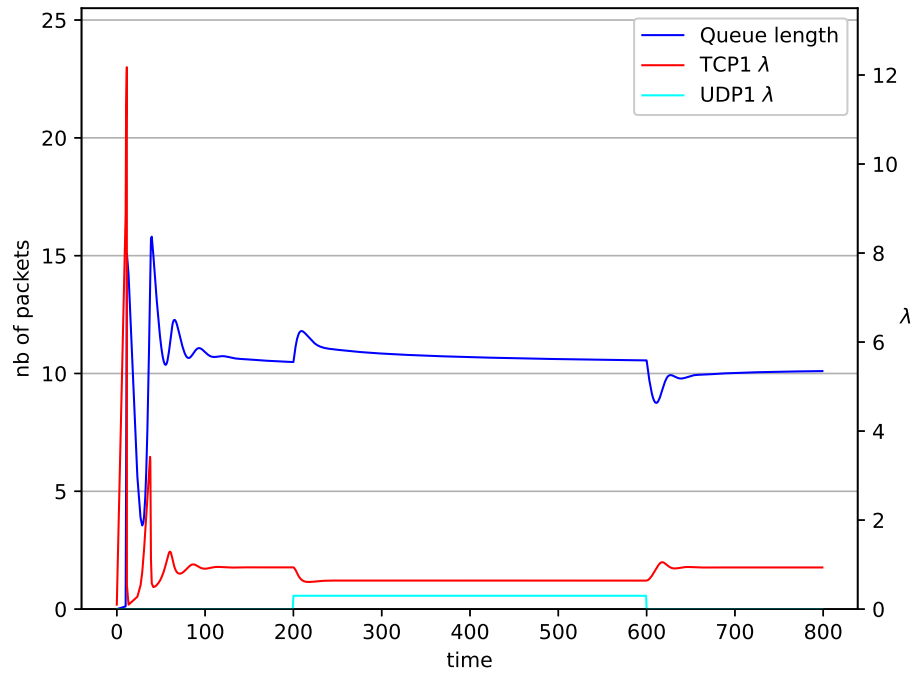


Figure 6. 1 TCP and 1 UDP with  $PI^\gamma$  controller.

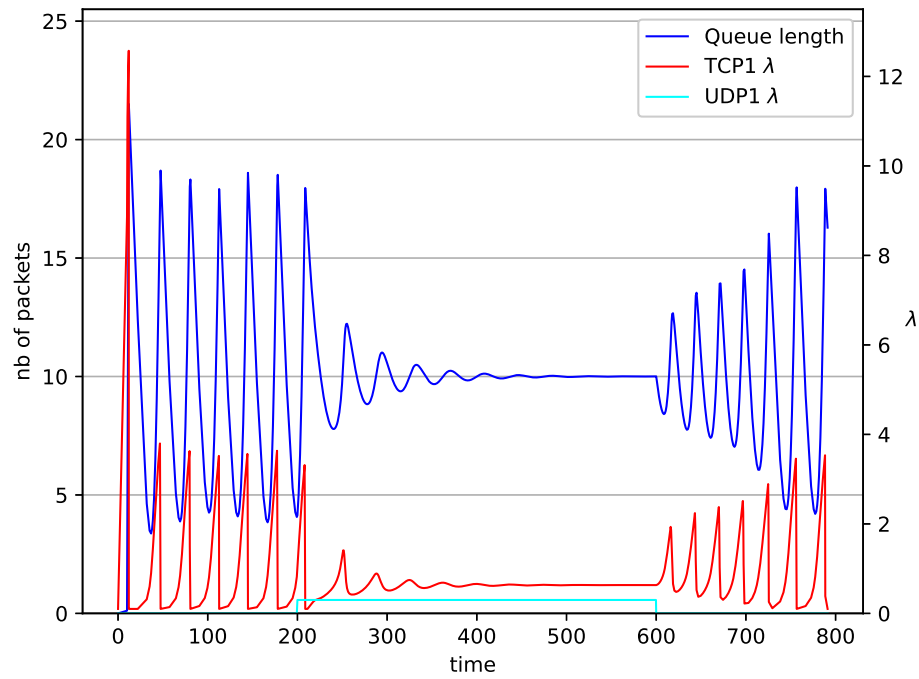


Figure 7. 1 TCP and 1 UDP with PI controller.

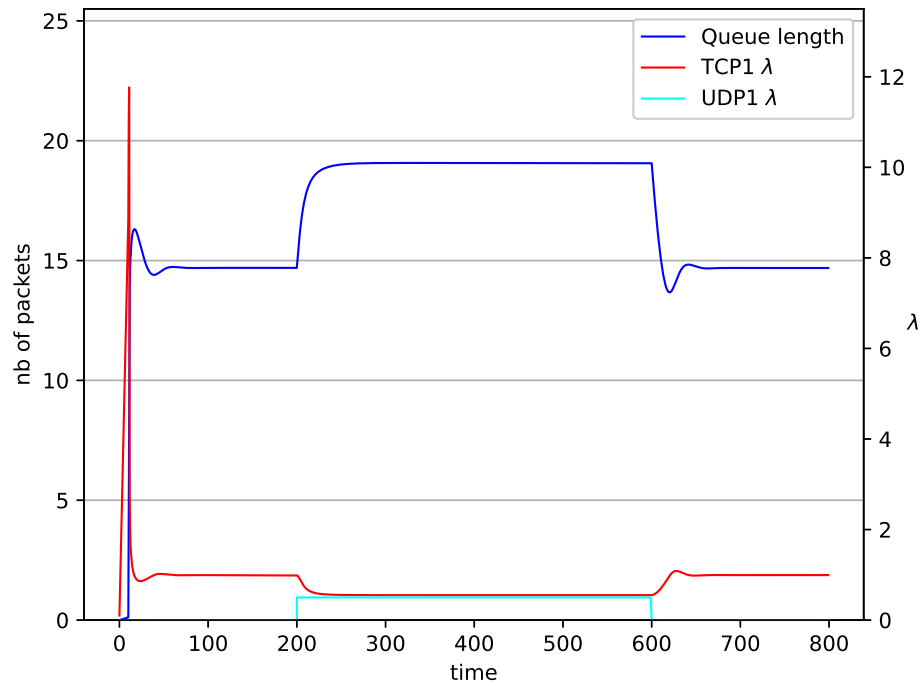


Figure 8. 1 TCP and 1 UDP with RED controller.

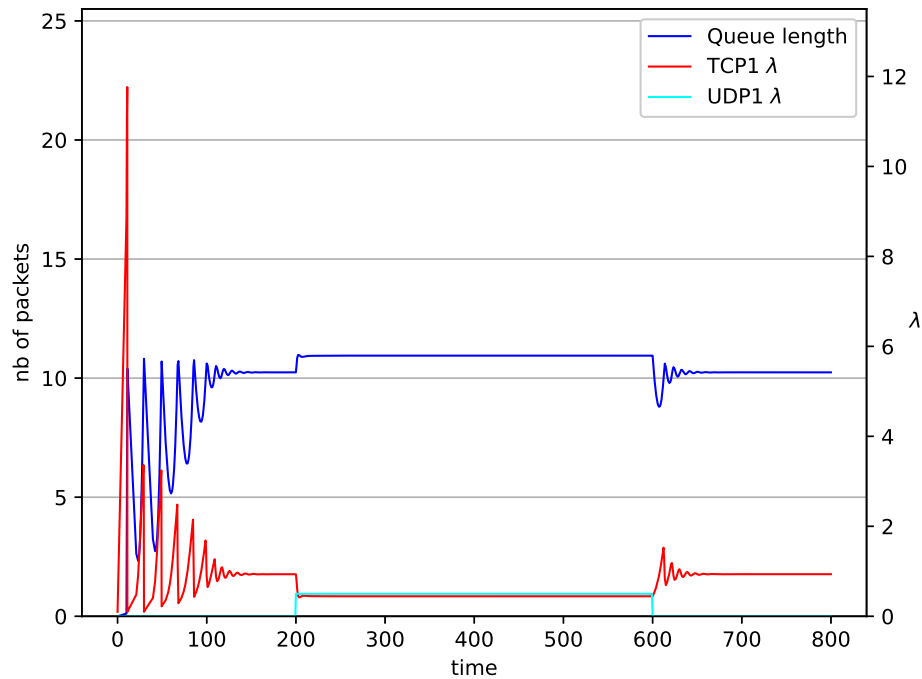


Figure 9. 1 TCP and 1 UDP with NLRED controller.

In Figures 10–13 the case of one TCP and two UDP streams employed is depicted. TCP works all the time. The first constant bit rate UDP stream ( $\lambda = 0.5$ ) starts transmission at time  $t = 200$  and finishes at  $t = 600$ . The second UDP stream ( $\lambda = 0.3$ ) transmits between  $t = 350$  and  $t = 450$ . The second UDP stream causes queue overload. In both cases ( $PI^\gamma$  and RED queue), the queue’s occupancy increases dramatically. The  $PI^\gamma$  mechanism slowly stabilizes the queue, reducing the queue size (Figure 11). The RED (Figure 12) and NLRED

(Figure 13) mechanisms cannot handle such amounts of input data. In the case of a RED algorithm, the size of the queue oscillates around the point  $Max_{th}$ .

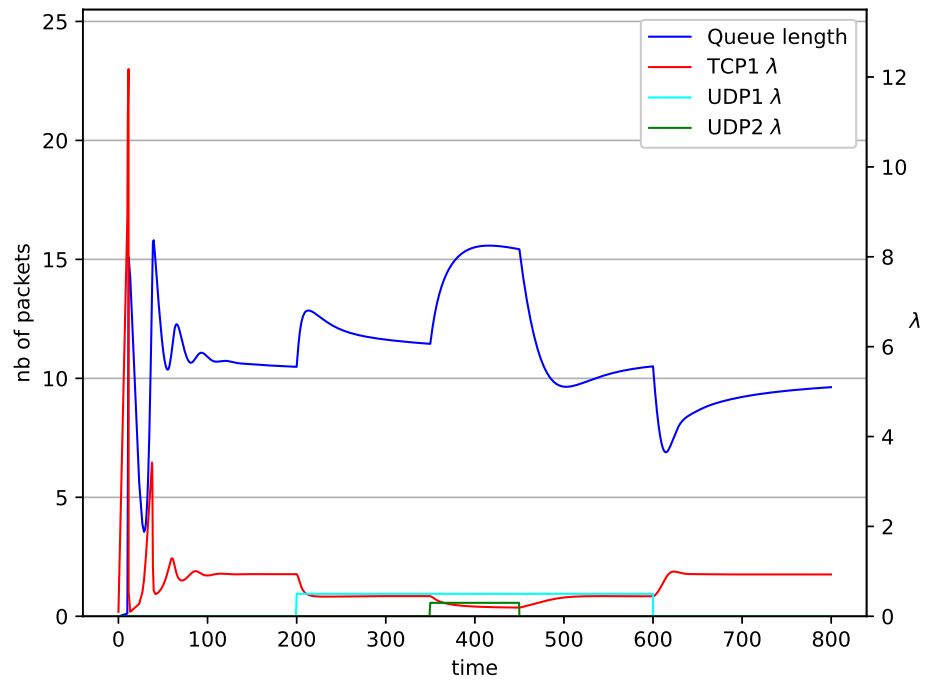


Figure 10. 1 TCP and 2 UDP with  $PI^\gamma$  controller.

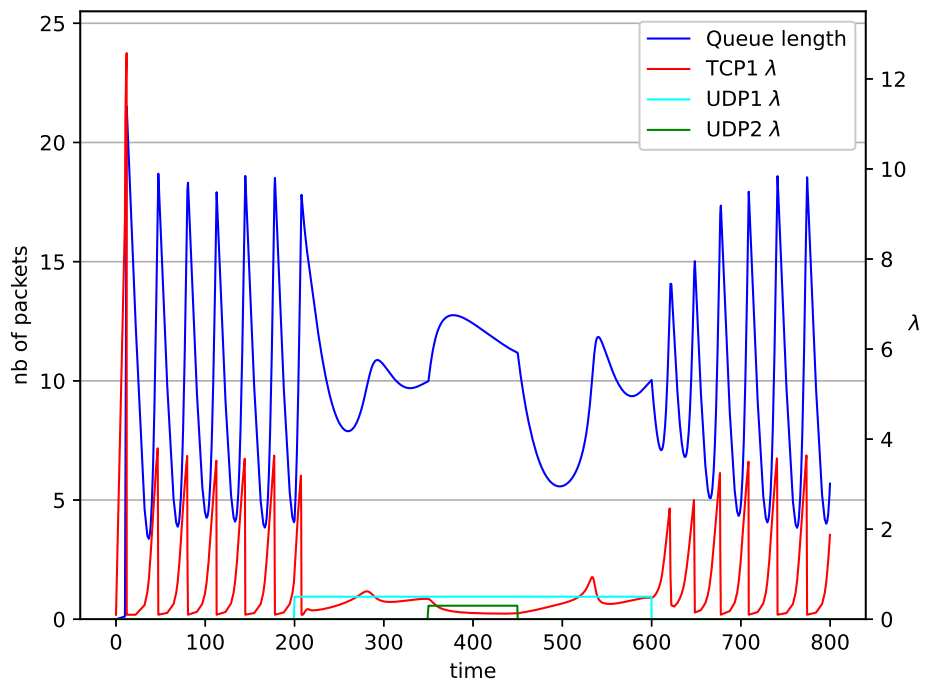


Figure 11. 1 TCP and 2 UDP with  $PI$  controller.

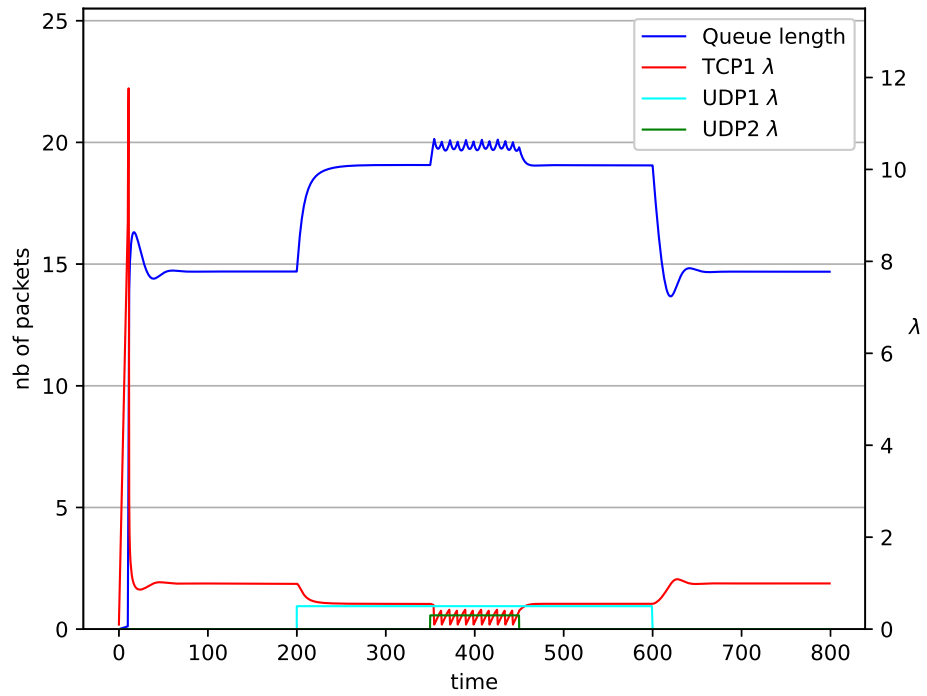


Figure 12. 1 TCP and 2 UDP with RED controller.

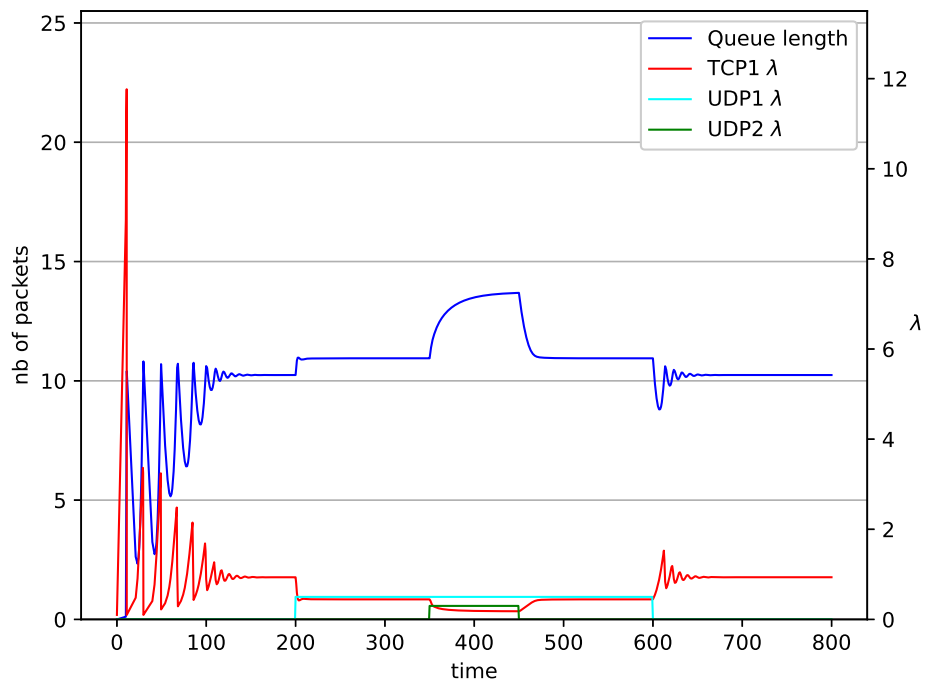


Figure 13. 1 TCP and 2 UDP with NLRED controller.

In Figures 14–17 the situation when two TCP streams cooperate is shown. At the moment when the second stream is switched on, the intensity of both sources stabilises at the same level ( $\lambda = 0.5$ ). In the case of the RED algorithm, the queues occupancies are set at values close to the  $Max_{th} = 19.45$ . For NLRED mechanism, the average queue size reaches 10.1. In the case of the  $PI^r$  controller, the queue length, after a temporary increase, sags to the desired value. This trend ends when the second source ends its transmission.

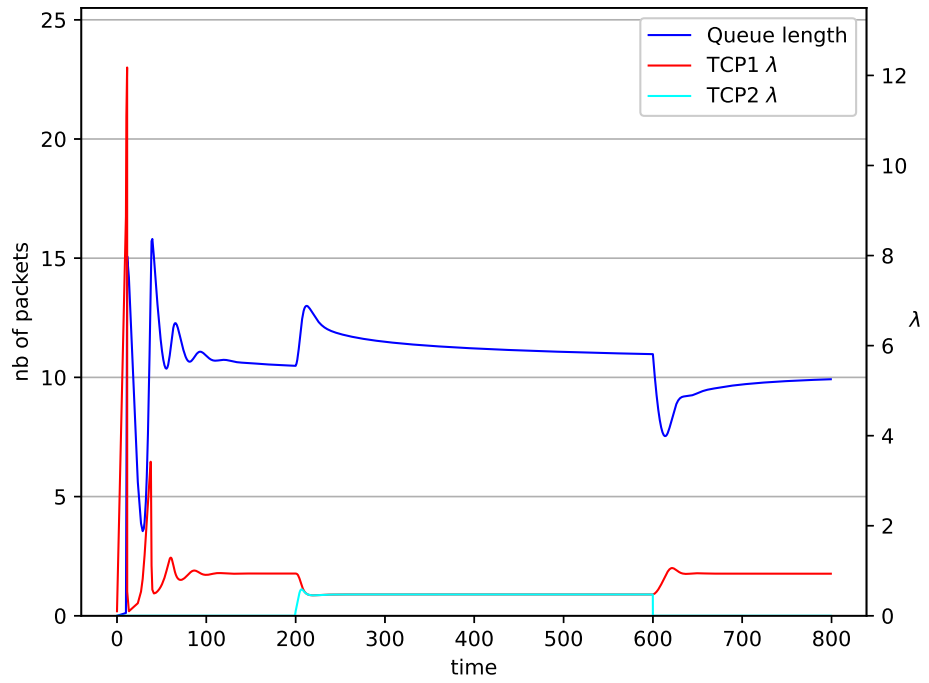


Figure 14. 2 TCP with  $PI^\gamma$  controller.

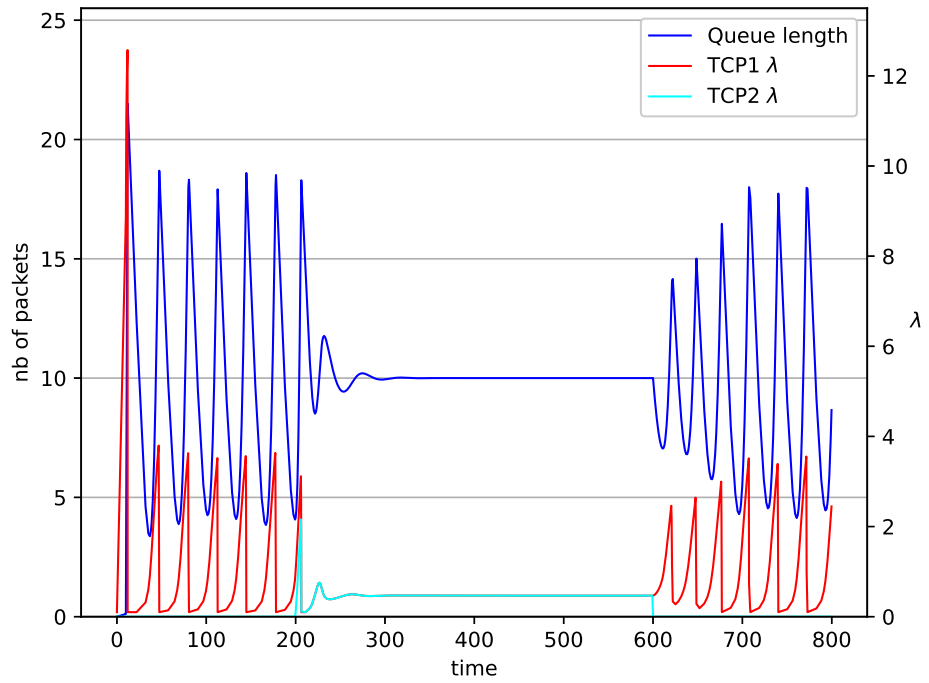


Figure 15. 2 TCP with PI controller.

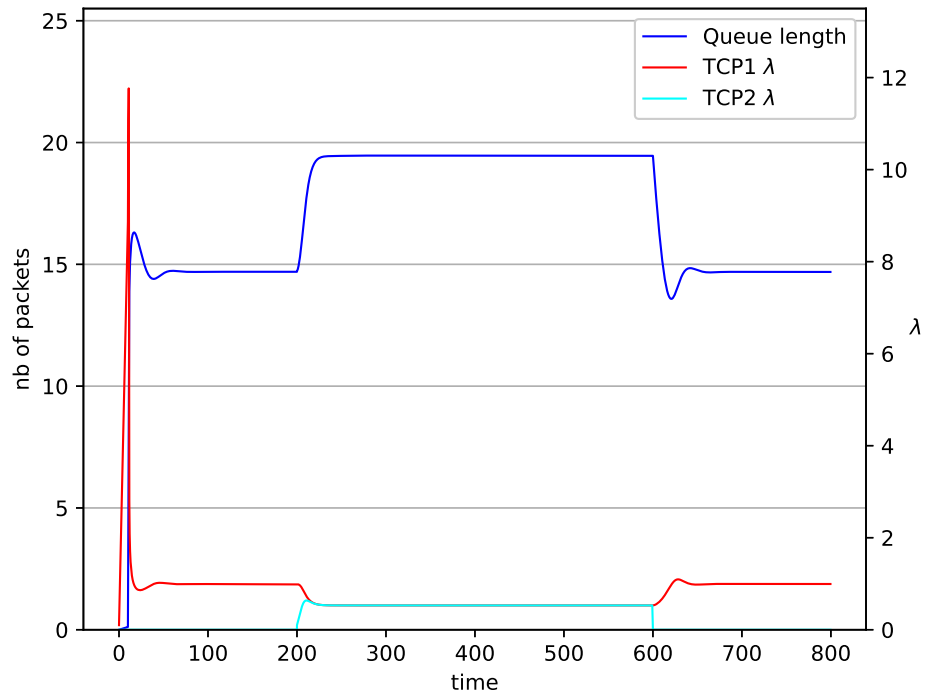


Figure 16. 2 TCP with RED controller.

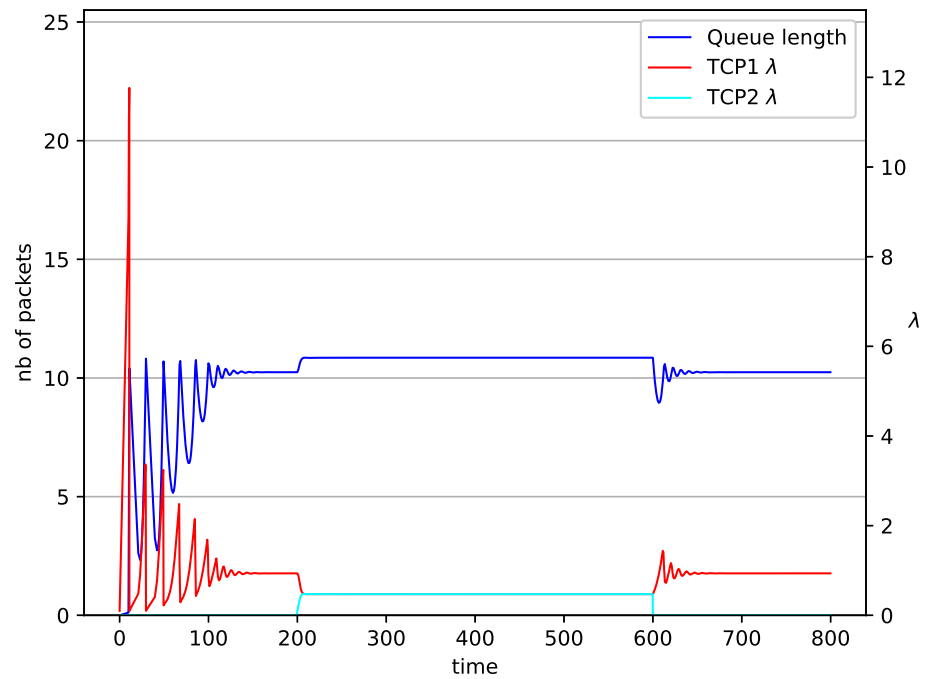


Figure 17. 2 TCP with NLRED controller.

### 6. Conclusions

This article describes the use of the diffusion approximation to estimate the behavior of the AQM mechanisms, their influence on the evolution of the TCP congestion window and potential cooperation of more TCP/UDP streams. Analytical methods for assessing network mechanisms known from the literature, with the exception of the fluid flow method, are not very suitable for evaluating TCP mechanisms. In the article [17], we have

presented a new approach of using the diffusion approximation to model the behavior of the TCP stream. In this article, we expand the given mechanism with the possibility of evaluation of cooperation of more TCP/UDP streams.

The experiments also show the good points of the fractional order controller  $PI^\gamma$ . Decreasing the integration order of the controller reduces the queue fluctuation.

On the other hand, the new approach gets a completely different view on the cooperation of the TCP/UDP protocol with the AQM mechanisms. Our model allows starting and ending the TCP/UDP transmission at any time. The described approach allows us to observe the dynamics of a transmission when different sources start or end their transmission.

The impact of the AQM algorithm on the dynamics of the TCP window is also presented. Our study shows differences in queue behavior in the case of different AQM mechanisms.

**Author Contributions:** Conceptualization, D.M., A.D., J.D., J.S., T.C. and J.K.; Formal analysis, J.D. and J.K.; Investigation, D.M. and J.S.; Methodology, A.D., D.M. and T.C.; Project administration, J.K.; Software, A.D., D.M. and J.S.; Supervision, A.D. and J.D.; Validation, T.C.; Visualization, D.M. and J.S.; Writing—original draft, A.D. and J.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was financed by National Science Center project no. 2017/27/B/ST6/00145.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Yue, Z.; Zhang, X.; Ren, Y.; Li, J.; Zhong, Q. The performance evaluation and comparison of TCP-based high-speed transport protocols. In Proceedings of the 2012 IEEE International Conference on Computer Science and Automation Engineering, Beijing, China, 22–24 June 2012; pp. 509–512.
2. Chydzinski, A. On the Transient Queue with the Dropping Function. *Entropy* **2020**, *22*, 825.
3. Grazia, C.A.; Patriciello, N.; Klapez, M.; Casoni, M. A cross-comparison between TCP and AQM algorithms: Which is the best couple for congestion control? In Proceedings of the 14th International Conference on Telecommunications (IEEE ConTEL), Zagreb, Croatia, 28–30 June 2017; pp. 75–82.
4. Floyd, S.; Jacobson, V. Random Early Detection gateways for congestion avoidance. *IEEE/ACM Trans. Netw.* **1993**, *1*, 397–413.
5. Feng, C.; Huang, L.; Xu, C.; Chang, Y. Congestion Control Scheme Performance Analysis Based on Nonlinear RED. *IEEE Syst. J.* **2017**, *11*, 2247–2254.
6. Kahe, G.; Jahangir, A.H. A self-tuning controller for queuing delay regulation in TCP/AQM networks. *Telecommun. Syst.* **2019**, *71*, 215–229.
7. Feng, W.-C.; Shin, K.G.; Kandlur, D.D.; Saha, D. The BLUE Active Queue Management algorithms. *IEEE/ACM Trans. Netw.* **2002**, *10*, 513–528.
8. Hu, L.; Kshemkalyani, A.D. HRED: A simple and efficient Active Queue Management algorithm. In Proceedings of the 13th International Conference on Computer Communications and Networks (IEEE Cat. No.04EX969), Chicago, IL, USA, 11–13 October 2004; pp. 387–393.
9. Long, C.; Zhao, B.; Guan, X.; Yang, J. The Yellow Active Queue Management algorithm. *Comput. Netw.* **2005**, *47*, 525–550.
10. Kelly, F. Mathematical Modelling of the Internet. In *Mathematics Unlimited—2001 and Beyond*; Springer: Berlin/Heisenberg, Germany, 2001; pp. 685–702.
11. Athuraliya, S.; Low, S.H.; Li, V.H.; Yin, Q. REM: Active Queue Management. *IEEE Netw.* **2001**, *15*, 48–53.
12. Li, Z.; Liu, Y.; Jing, Y. Active Queue Management Algorithm for TCP Networks with Integral Backstepping and Minimax. *Int. J. Control Autom. Syst.* **2019**, *17*, 1059–1066.
13. Hollot, C.V.; Misra, V.; Towsley, D. Analysis and design of controllers for AQM routers supporting TCP flows. *IEEE Trans. Autom. Control* **2002**, *47*, 945–959.
14. Shah, P.; Yasmin, S.; Asghar, S.; Qayyum, A.; Hasbullah, H. A Fluid Flow Model for SCTP Traffic over the Internet. In Proceedings of the International Conference on Emerging Technologies (ICET), Islamabad, Pakistan, 8–9 October 2012; pp. 1–6.
15. Domańska, J.; Domański, A.; Czachórski, T.; Klamka, J. Fluid flow approximation of time-limited TCP/UDP/XCP streams. *Bull. Pol. Acad. Sci. Tech. Sci.* **2014**, *62*, 217–225.
16. Bonaventura, M.; Castro, R. Fluid-flow and packet-level models of data networks unified under a modular/hierarchical framework: Speedups and simplicity, combined. In Proceedings of the 2018 Winter Simulation Conference (WSC), Gothenburg, Sweden, 9–12 December 2018; pp. 3825–3836.

17. Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Szyguła, J.; Marek, D. Diffusion Approximation Model of TCP NewReno Congestion Control Mechanism. *Springer Nat. Comput. Sci.* **2020**, *1*, 43.
18. Unal, H.; Melchor-Aguilar, D.; Ustebay, D.; Niculescu, S.I.; Ozbay, H. Comparison of PI controllers designed for the delay model of TCP/AQM. *Comput. Commun.* **2013**, *36*, 1225–1234.
19. Fan, Y.; Ren, F.; Lin, C. Design a PID controller for Active Queue Management. In Proceedings of the Eighth IEEE Symposium on Computers and Communications, ISCC 2003, Kemer-Antalya, Turkey, 3 July 2003; Volume 2, pp. 985–990.
20. Krajewski, W.; Viaro, U. On robust fractional order PI controller for TCP packet flow. In Proceedings of the BOS Conference: Systems and Operational Research, Angers, France, 6–8 March 2014.
21. Domańska, J.; Domański, A.; Czachórski, T.; Klamka, J. The use of a non-integer order PI controller with an Active Queue Management Mechanism. *Int. J. Appl. Math. Comput. Sci.* **2016**, *26*, 777–789.
22. Domańska, J.; Domański, A.; Czachórski, T.; Klamka, J. Self-similarity Traffic and AQM Mechanism Based on Non-integer Order  $PI^\alpha D^\beta$  Controller. *Commun. Comput. Inf. Sci. Springer Int. Publ.* **2017**, *718*, 336–350.
23. Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Marek, D.; Szyguła, J. The Influence of the Traffic Self-similarity on the Choice of the Non-integer Order  $PI^\alpha$  Controller Parameters. *Commun. Comput. Inf. Sci. Springer Int. Publ.* **2018**, *935*, 76–83.
24. Szyguła, J.; Domański, A.; Domańska, J.; Czachórski, T.; Marek, D.; Klamka, J. AQM Mechanism with Neuron Tuning Parameters. In *Intelligent Information and Database Systems*; Springer: Berlin/Heisenberg, Germany, 2020; pp. 299–311.
25. Marek, D.; Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Szyguła, J. Combined diffusion approximation–simulation model of AQM’s transient behavior. *Elsevier Comput. Commun.* **2020**, *166*, 40–48.
26. Seferoglu, H.; Modiano, E. TCP-aware backpressure routing and scheduling. In Proceedings of the Information Theory and Applications Workshop (ITA), San Diego, CA, USA, 9–14 February 2014; pp. 1–9. doi:10.1109/ITA.2014.6804278.
27. Bohloulzadeh, A.; Mehri, R. A Survey on Congestion Control Protocols in Wireless Sensor Networks. *Int. J. Wirel. Inf. Netw.* **2020**, *365*–384.
28. Athanasopoulou, E.; Bui, L.X.; Ji, T.; Srikant, R.; Stolyar, A. Back-Pressure-Based Packet-by-Packet Adaptive Routing in Communication Networks. *IEEE/ACM Trans. Netw.* **2013**, *21*, 244–257.
29. Jan, M.A.; Jan, S.; Alam, M.; Akhuzada, A.; Rahman, I. A Comprehensive Analysis of Congestion Control Protocols in Wireless Sensor Networks. *Mob. Netw. Appl.* **2018**, *23*, 456–468.
30. Misra, V.; Gong, W.; Towsley, D. Fluid-based analysis of network of AQM routers supporting TCP flows with an application to RED. *Comput. Commun. Rev.* **2000**, *30*, 151–160.
31. Hong, Y.; Yang, O.W.W. Adaptive AQM controllers for IP routers with a heuristic monitor on TCP flows. *Int. J. Commun. Syst.* **2006**, *19*, 17–38.
32. Sun, J.; Ko, K.; Chen, G.; Chan, S.; Zukerman, M. PD-RED: To improve the performance of RED. *IEEE Commun. Lett.* **2003**, *7*, 406–408.
33. Chen, Y.; Petras, I.; Xue, D. Fractional order control—A tutorial. In Proceedings of the American Control Conference, St. Louis, MO, USA, 10–12 June 2009; pp. 1397–1411.
34. Domańska, J.; Domański, A.; Czachórski, T.; Klamka, J.; Szyguła, J. The AQM Dropping Packet Probability Function Based on Non-integer Order  $PI^\alpha D^\beta$  Controller. In *Lecture Notes in Electrical Engineering*; Springer: Berlin/Heisenberg, Germany, 2019; Volume 496, pp. 36–48.
35. Czachórski, T.; Pekergin, F. Diffusion Approximation as a Modelling Tool. In *Network Performance Engineering—A Handbook on Convergent Multi-Service Networks and Next Generation Internet, LNCS*; Springer: Berlin/Heisenberg, Germany, 2011; Volume 5233, pp. 447–476.
36. Czachórski, T.; Grochla, K.; Nycz, T.; Pekergin, F. A diffusion approximation model for wireless networks based on IEEE 802.11 standard. *Comput. Commun.* **2010**, *33*, 86–92.
37. Nycz, T.; Nycz, M.; Czachórski, T. A Numerical Comparison of Diffusion and Fluid-Flow Approximations Used in Modelling Transient States of TCP/IP Networks. *Commun. Comput. Inf. Sci.* **2014**, *431*, 213–222.
38. Qi, Y.; Ohtsuki, T. QoE-Aware Video Streaming Transmission Optimization Method for Playout Threshold Adjustment in LTE. In Proceedings of the 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), Porto, Portugal, 3–6 June 2018; pp. 1–5.
39. Zhou, K.; Yeung, K.; Li, V. Nonlinear RED: A simple yet efficient Active Queue Management scheme. *Comput. Netw. Int. J. Comput. Telecommun. Netw.* **2006**, *50*, 3784–3794.
40. Domańska, J.; Augustyn, D.; Domański, A. The choice of optimal 3-rd order polynomial packet dropping function for NLRED in the presence of self-similar traffic. *Bull. Pol. Acad. Sci. Tech. Sci.* **2012**, *60*, 779–786.
41. Podlubny, I. Fractional order systems and  $PI^\lambda D^\mu$  controllers. *IEEE Trans. Autom. Control* **1999**, *44*, 208–214.
42. Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Marek, D.; Szyguła, J. GPU Accelerated Non-integer Order  $PI^\alpha D^\beta$  Controller Used as AQM Mechanism. In *Communications in Computer and Information Science*; Springer: Berlin/Heisenberg, Germany, 2018; Volume 860, pp. 286–299.
43. Miller, K.; Ross, B. *An Introduction to the Fractional Calculus and Fractional Differential Equations*; Wiley: New York, NY, USA, 1993.
44. Podlubny, I. *Fractional Differential Equations*; Academic Press: San Diego, CA, USA, 1999; Volume 198.
45. Ciesielski, M.; Leszczynski, J. A Numerical Method for Solution of Ordinary Differential Equations of Fractional Order. In *Parallel Processing and Applied Mathematics*; Springer: Berlin/Heisenberg, Germany, 2002; Volume 2328, pp. 695–702.
46. Newell, G.F. Queues with time-dependent arrival rates. I — The transition through saturation. *J. Appl. Probab.* **1968**, *2*, 436–451.



47. Gelenbe, E. On Approximate Computer Systems Models. *J. ACM* **1975**, *22*, 261–269.
48. Gelenbe, E. A Diffusion Model for Packet Travel Time in a Random Multi-hop Medium. *ACM Trans. Sens. Netw. (TOSN)* **2007**, *3*, 10.
49. Czachórski, T. A method to solve Diffusion Equation with Instantaneous return Processes Acting as Boundary Conditions. *Bull. Pol. Acad. Sci. Tech. Sci.* **1993**, *41*, 417–451.
50. Bonald, T.; May, M.; Bolot, J. Analytic evaluation of RED performance. In Proceedings of the INFOCOM, Tel Aviv, Israel, 26–30 March 2000.
51. Zheng, B.; Atiquzzaman, M. A Framework to Determine the Optimal Weight Parameter of RED in Next-Generation Internet Routers. *Int. J. Commun. Syst.* **2008**, *21*, 987–1008.

## **Rozdział 10**

# **Approximation Models for the Evaluation of TCP/AQM Networks**

# Approximation models for the evaluation of TCP/AQM networks

Dariusz MAREK<sup>1</sup>, Adam DOMAŃSKI<sup>1</sup>, Joanna DOMAŃSKA<sup>2</sup>, Jakub SZYGULA<sup>1</sup>\*,  
Tadeusz CZACHÓRSKI<sup>2</sup>, Jerzy KLAMKA<sup>2</sup>, and Katarzyna FILUS<sup>2</sup>

<sup>1</sup> Faculty of Automatic Control, Electronics and Computer Science, Department of Distributed Systems and Informatic Devices, Silesian University of Technology, Address: Akademicka 16, 44-100 Gliwice, Poland

<sup>2</sup> Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland

**Abstract.** The article proposes a model in which Diffusion Approximation is used to analyse the TCP/AQM transmission mechanism in a multi-node computer network. In order to prevent traffic congestion, routers implement AQM (Active Queue Management) algorithms. We investigate the influence of using RED-based AQM mechanisms and the fractional controller  $PI^\gamma$  on the transport layer. Additionally, we examine the cases in which the TCP and the UDP flows occur and analyse their mutual influence. Both transport protocols used are independent and work simultaneously. We compare our solution with the Fluid Flow approximation, demonstrating the advantages of Diffusion Approximation.

**Key words:** Diffusion Approximation; Active Queue Management; congestion control; dropping packets; Fluid Flow Approximation; non-integer order  $PI^\gamma$  controller; G/G/1/N queueing model.

## 1. INTRODUCTION

The purpose of communication protocols used in computer networks is to ensure efficient and correct transmission between devices. It can be done in many ways. This article focuses on a specific group of mechanisms increasing transmission efficiency by minimising possible packet losses due to buffer overflows at transmission nodes. The Internet Engineering Task Force (IETF) suggests using Active Queue Management (AQM) algorithms for this purpose. These algorithms allow to preemptively drop packets, even if the buffer is not full, but there is a risk of overflowing it soon. A significant increase in Internet traffic and poor congestion control can result in partial or complete degradation of network performance [1]. Therefore, numerous works discuss Active Queue Management [1, 2] as a remedy. Malicious actors can cause the increase in traffic intensity via Distributed Denial of Service, which becomes an increasingly frequent problem of the modern Internet [3]. Therefore, the analysis and modelling of TCP/AQM are also interesting for cybersecurity.

Since the creation of the primary AQM mechanism (Random Early Detection, RED), many modifications have been proposed, e.g. BLUE algorithm [4], hyperbola RED (HRED) [5], and Yellow algorithm [6]. The alternatives to RED algorithms can be, e.g. CoDel and PIE [7]. However, although many years have passed since the RED mechanism was introduced, this mechanism is still the most popular AQM scheme [8, 9]; therefore it can serve as a baseline AQM.

\*e-mail: jakub.szygula@polsl.pl

Manuscript submitted 2022-03-18, revised 2022-05-30, initially accepted for publication 2022-06-26, published in August 2022.

The mechanism of RED algorithms is based on a packet loss probability function. The loss probability generally depends on the queue moving average of the length and affects the number of packets to be dropped. In most cases, increasing queue occupancy increases the probability of incoming packet drop. Different probability functions can be used. In the case of the RED algorithm, [10] it is a linear function. For NLRED algorithm [11, 12] it can be a quadratic or polynomial function. For more sophisticated cases, e.g. [13], the probability of packet drop can be calculated based on the PI controller response.

One of the advantages of AQM algorithms is that they cooperate with the transport layer protocol – TCP, which has its built-in congestion window control mechanisms. When possible, these mechanisms increase the transmission rate and decrease it when a data segment is lost. There are many congestion window control algorithms [14]. The TCP NewReno [15] algorithm is the most widely used in Internet transmissions. Random packet deletion provided by AQM allows avoiding so-called global synchronisation effect, which involves slowing down transmission by multiple TCP transmitters at the same time in case of losses caused by exceeding maximum queue size in nodes. The combination of these both mechanisms: AQM determining packet losses in IP routers and TCP congestion window reacting on the loss, allows us to eliminate congestion effectively in the network [16].

A dynamic TCP model is necessary to reflect the time-dependent behaviour of flows and queues. The most commonly used and the simplest mechanism to model TCP protocol behavior in cooperation with AQM is the Fluid Flow approximation [17–20]. As an alternative, simulation models can also be used. However, they are very time-consuming in the case of

transient-state analysis, crucial in the evaluation of Internet behaviour [21].

In this paper, we present and use a Diffusion Approximation network model to evaluate network transmission efficiency based on the two mechanisms presented: a TCP NewReno-based transmission rate control mechanism and a preemptive packet deletion mechanism. The model presented in this paper is a multi-node extension of our earlier models presented in [22].

The Diffusion Approximation is a second-order approximation (it takes into account two first moments of interarrival and service times distributions), and it provides the distributions of queue lengths and waiting times; therefore it is more accurate than Fluid Flow Approximation, based only on mean values.

To create a diffusion TCP/AQM network model, we used the generic network models proposed in [23–25] having no control mechanisms. Three AQM algorithms were selected for the study: the basic RED algorithm and two of its modifications: NLRED and  $PI^\lambda$ .

The rest of the paper is organised as follows. In Section 2 one can find the literature overview regarding AQM techniques based on the control theory methods and an embedding of our solution in the area of diffusion modelling. Section 3 gives a brief description of the AQM techniques used in this paper. Section 4 describes our diffusion model of the non-integer  $PI^\gamma$  controller with TCP/UDP streams. In Section 5 numerical results are presented. Section 6 concludes the article.

## 2. BACKGROUND AND RELATED WORK

Congestion management is one of the critical issues in the domain of computer networks [16]. The most effective solutions are based on the TCP congestion window (CWND) management mechanism [26]. For UDP-based transmissions, mechanisms based on choke packets can be used [27]. These mechanisms are adequate for bandwidth control; however, they introduce transmission delays. Despite their disadvantages, they are still used, especially in wireless sensor networks (WSNs), due to their limited computational resources, storage, energy, and communication bandwidth requirements [27, 28]. The use of choke packets in TCP networks may be ineffective due to incompatibility with the backpressure queue size based routing [26]. The possible cooperation of TCP and backpressure mechanisms would require changes in the management mechanism of CWND in the TCP protocol [26].

AQM mechanisms are well adapted to work with the TCP protocol. Their cooperation can be modelled as a closed-loop control algorithm with AQM as a controller. Control theory traditional methods may be applied to investigate and improve proposed solutions stability, efficiency, and performance.

Paper [29] introduces a Fluid Flow dynamic model of TCP/RED based on stochastic differential equations. The presented model allowed modelling of the queue behaviour and the CWND mechanism for the TCP protocol. The traffic having both the TCP and the UDP streams has been considered in [30]. One of the properties of this model was the continuous merging of CBR/UDP traffic with the TCP stream. The paper [19] intro-

duced a Fluid Flow model that allows TCP and UDP streams to be treated independently. The streams for this model are also time-constrained. In the paper [31] a Fluid Flow model considering multiple AQM nodes was presented; [32] extends it to a vast network with topology mapped from the real Internet.

The control theory has resulted in the emergence of entirely new AQM algorithms based on the response from a PID controller or one of its variants. The use of a Proportional-Integral (PI) controller for low-frequency dynamics was proposed in the paper [17]. The article [33] makes a comparison of adaptive Proportional (P) and Proportional-Integral (PI) controllers; the PI controller was found to be more adaptive to large fluctuations in network traffic. A new algorithm from the RED family was proposed in [34]. The proposed Proportional-Derivative-RED (PD-RED) mechanism demonstrated better adaptability to network traffic than the previously known Adaptive RED algorithm. The acceleration of the controller's response to changing network conditions was achieved using the Proportional-Integral-Differential (PID) algorithm [35]. Generally, the advantage of PID controllers is their computation, and implementation simplicity [36]. An attempt to maintain this simplicity in more complex mechanisms has been described in the article [37]. The authors present a self-tuning compensated PID controller.

All the above articles are based on traditional integer-order differentiation and integration. As shown in [38], the behaviour of many real-world dynamic systems can be modelled more accurately using a fractional dynamic model. Such models provide better performance than the conventional integer-order ones. The first attempt to use the fractional-order PI controller in an AQM strategy was proposed in the article [39]. However, the author only focused on determining the range where the  $PI^\gamma$  controller ensures a given modulus margin (inverse of the  $H_\infty$  norm of the sensitivity function). A fully working AQM model based on  $PI^\gamma$  controller was proposed in [40], where Fluid Flow approximation and simulation models were used to demonstrate the correctness of the proposed mechanism. A method of correct selection of  $PI^\alpha D^\beta$  controller parameters with the use of simulations was described in [41]. The article [22] presents a new model of the  $PI^\gamma$  controller based on a Diffusion Approximation. The article proves that the diffusion model allows us to obtain more detailed information on transmission delays and more accurately reflects real network traffic than the Fluid Flow model.

The tutorial [42] shows how diffusion is used in the analysis of traffic control mechanisms implemented in ATM networks. The use of the model in wireless networks has been described in [43]. TCP/RED mechanism has been first modelled in the article [44]. The CWND TCP mechanism and its ability to interoperate with various AQMs were described in [22]. A combined diffusion and simulation model for the analysis of TCP/AQM mechanisms has been proposed in [45]. A network traffic model based on independent TCP and UDP streams has been proposed in [46].

The models presented above can be used to analyse how Internet traffic (understood as a collection of TCP and UDP streams) affects the behaviour of the AQM queue in a sin-

gle network node. In this paper, we present a TCP/UDP/AQM model that reflects the behaviour of a more significant number of routers. The flexibility of the proposed model allows us to create network models with various and arbitrarily complex topologies and obtain more accurate results than in the case of Fluid models. To the best of our knowledge, no diffusion model proposed in the literature can be used for the TCP/UDP/AQM networks analysis.

### 3. AQM MECHANISMS BASED ON PACKET DROPPING PROBABILITY FUNCTION

The rate control mechanisms built into the TCP protocol are based on a packet loss mechanism. In case of loss, the transmitter decreases the transmission rate by reducing the size of the congestion window. If valid acknowledgments of the transmission are received, the transmission rate is increased. In static queue management, packet loss is a consequence of buffer overflow. In AQM mechanisms, packets can be dropped earlier, even when the queue is not yet overflowing. This drop is random and usually depends on the queue length. In our paper, we use three different mechanisms that differ in the calculation of packet loss probability: RED, NLRED, and  $PI^\gamma$ .

The RED family algorithms (which include RED and NLRED) use the weighted moving average  $\text{avg}$  computed at packets' arrival; for packet  $i$ :

$$\text{avg}_i = (1 - w)\text{avg}_{i-1} + wq_i,$$

where  $\text{avg}_{i-1}$  is used to denote the moving average determined at the arrival of the previous packet, whereas  $q_i$  denotes the queue length.

As  $\text{avg}$  value increases, the probability grows linearly between two thresholds  $\text{Min}_{th}$  and  $\text{Max}_{th}$ , from 0 to  $p_{\max}$ .

$$P_{\text{RED}}(\text{avg}) = p_{\max} \frac{\text{avg} - \text{Min}_{th}}{\text{Max}_{th} - \text{Min}_{th}}. \quad (1)$$

Finally, it becomes  $p_{\text{RED}} = 1$  for  $\text{avg} > \text{Max}_{th}$ .

In case of NLRED (non-linear RED) algorithm, the linear packet drop function is replaced by a polynomial function [11]. The packet dropping probability function is based on the third-degree polynomials:

$$p(\text{avg}, a_1, a_2, p_{\max}) = \begin{cases} 0 & \text{for } \text{avg} < \text{Min}_{th}, \\ \varphi_0(\text{avg}) + a_1 \varphi_1(\text{avg}) & \text{for } \text{Min}_{th} \leq \text{avg} \leq \text{Max}_{th}, \\ 1 & \text{for } \text{avg} > \text{Max}_{th}, \end{cases} \quad (2)$$

where  $a_1, a_2$  are NLRED coefficients and:

$$\varphi_0(\text{avg}) = p_{\max} \frac{\text{avg} - \text{Min}_{th}}{\text{Max}_{th} - \text{Min}_{th}}, \quad (3)$$

$$\varphi_1(\text{avg}) = (\text{avg} - \text{Min}_{th})(\text{Max}_{th} - \text{avg}), \quad (4)$$

$$\varphi_2(\text{avg}) = (\text{avg} - \text{Min}_{th})^2 (\text{Max}_{th} - \text{avg}). \quad (5)$$

In the case of the third solution  $PI^\gamma$  probability  $p$  is determined based on a comparison of the current queue  $q_i$  and the queue  $q_0$  we want to maintain. This difference, called "error" in the control theory, affects the controller response, treated as packet dropping probability. In our case, we use the fractional-order proportional-integral  $PI^\gamma$  controller [13].

In this case the loss probability  $p_i$  of a packet  $i$  equals:

$$p_i = \max \{0, -(K_P e_i + K_I \Delta^\gamma e_i)\}. \quad (6)$$

This probability depends on  $K_P, K_I$  (the proportional and integral term respectively), the order of integration  $\gamma$  and the error  $e_i = q_i - q$ , where:  $e_i$  is an error in current slot,  $q_i$  is actual queue length and  $q$  (setpoint) is desired queue length. Articles [40, 41, 47, 48] discuss the selection and impact of these variables on controller effectiveness.

The method of non-integer order discrete differ-integral calculation is analogous to the generalization employed in the Grünwald-Letnikov (GrLET) formula [49, 50].

For a given sequence  $f_0, f_1, \dots, f_j, \dots, f_k$

$$\Delta^\gamma f_k = \sum_{j=0}^k (-1)^j \binom{\gamma}{j} f_{k-j}, \quad (7)$$

where  $\gamma \in R$  is usually a non-integer fractional order,  $f_k$  is a differentiated discrete function, and  $\binom{\gamma}{j}$  is a generalised Newton symbol which can be defined as follows:

$$\binom{\gamma}{j} = \begin{cases} 1 & \text{for } j = 0, \\ \frac{\gamma(\gamma-1)(\gamma-2)\dots(\gamma-j+1)}{j!} & \text{for } j = 1, 2, \dots \end{cases} \quad (8)$$

### 4. THE APPROXIMATION MODELS OF TCP/AQM BEHAVIOUR

This section describes the basics of the Fluid Flow and Diffusion Approximation models and how they can model TCP/AQM mechanisms. Network extensions to both models will also be described to evaluate AQM queues over multiple transmission nodes.

#### 4.1. Fluid Flow model

The Fluid Flow model was first presented in [29]. This model demonstrates TCP protocol dynamics in a simplified way (the model ignores the TCP timeout mechanisms). It is based on the following nonlinear differential equation describing the evolution of the congestion window size [51]:

$$\frac{dW_i^{TCP}(t)}{dt} = \frac{1}{R_i(t)} - \frac{W_i^{TCP}(t) W_i^{TCP}(t - R(t))}{2 R_i(t - R_i(t))} p(t - R_i(t)) \quad (9)$$

together with the balance equation for the congested router:

$$\frac{dq(t)}{dt} = \sum_{i=1}^N \frac{W_i^{TCP}(t)}{R_i(t)} - C, \quad (10)$$

where:

$W_i$  – expected size of the TCP congestion window (in packets) for a flow  $i$ , it defines the number of packets that may be sent without waiting for the acknowledgements of previous packets reception,

$R_i$  – round-trip time, can be calculated as  $R_i = q/C + T_p$ , whereas the sum  $\sum W_i/R_i$  denotes the total input flow to the examined congested router,

$q$  – queue length (in packets),

$C$  – link capacity (packets/time unit), the constant output flow of the router,

$T_p$  – propagation delay,

$N$  – number of TCP sessions passing through the router,

$p$  – packet dropping probability.

Model of UDP streams illustrates a CBR stream with assumed number of packet being sent per time unit. We refer to such a traffic because it is the most demanding traffic for the data link. The sending rate of the  $i$ -th UDP stream is approximated by the following equation:

$$W_i^{UDP}(t) = U. \quad (11)$$

The maximum values of  $q$  and  $W$  are dependent on the buffer capacity and maximum size of the window. Such a Fluid Flow model may be extended to any network topology with any number of TCP flows [52]. This method allows us to evaluate the behaviour of TCP/AQM mechanisms for large networks. The article [32] presents a numerical example for a network topology with 134 023 nodes and 50 000 flows.

#### 4.2. Diffusion Approximations

Diffusion Approximation can be applied in the queueing theory (e.g. [24, 53–55]) to determine a queue distribution. In this approximation method the discrete process representing a queue length is replaced by a continuous diffusion process  $X(t)$ . The probability density function (pdf) of this process  $f(x, t; x_0)$  [54]:

$$f(x, t; x_0) dx = P[x \leq X(t) < x + dx | X(0) = x_0] \quad (12)$$

can be defined as follows [54]:

$$\frac{\partial f(x, t; x_0)}{\partial t} = \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x}. \quad (13)$$

Equation (13) is used to evaluate the queue distribution in the G/G/1 queueing system, i.e. the one with general distributions of interarrival and service times. It is also used to analyse G/G/1/L systems, in which the queue is limited to a finite number  $L$  positions. As the number of packets is in the range  $[0, L]$ , the examined diffusion process is limited by the barriers at  $x = 0$  and  $x = L$ . The diffusion equation for this case supplemented by

balance equations for the barriers [54]:

$$\begin{aligned} \frac{\partial f(x, t; x_0)}{\partial t} &= \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x} \\ &\quad + \lambda p_0(t) \delta(x-1) + \lambda p_L(t) \delta(x-L+1), \\ \frac{dp_0(t)}{dt} &= \lim_{x \rightarrow 0} \left[ \frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0) \right] - \lambda p_0(t), \\ \frac{dp_L(t)}{dt} &= \lim_{x \rightarrow L} \left[ \frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0) \right] - \mu p_L(t). \end{aligned} \quad (14)$$

In the above equations,  $p_0(t)$  and  $p_L(t)$  denote probabilities that the process is in one of the barriers, the intensities of jumps from barriers (from  $x = 0$  to  $x = 1$  – an arrival of a packet to the empty queue – and from  $x = L$  to  $x = L - 1$  – a departure of a packet from the full queue) are denoted as  $\lambda$  and  $\mu$  respectively. Parameters  $\beta$  and  $\alpha$  are chosen as  $\beta = \lambda - \mu$ ,  $\alpha = \lambda^3 \sigma_A^2 + \mu^3 \sigma_B^2$ , where  $1/\lambda$  and  $1/\mu$  are the first moments of interarrival and service time distributions, and  $\sigma_A^2$ ,  $\sigma_B^2$  are their variances. This way the diffusion process and the queueing have the same mean and variance. The steady-state solution of equation (14) can be found in work [54] and the transient case is considered in [23]. The diffusion approximation model considering multiple independent data streams, including independent TCP and UDP streams, is described in [46].

The TCP/UDP/AQM diffusion model below allows us to obtain the mean value of queue length. This value is used to calculate the probability of packet loss and the new value of the congestion window. It is assumed that the flow  $\lambda$ , the delay  $q/\mu$  of the router, and the congestion window size  $W$  are linked in the following way:

$$\lambda = \frac{W\mu}{q}, \quad (15)$$

where  $1/\mu$  is the mean service time.

A queue distribution is obtained at time  $t_i$  (at the end of the  $i$ -th interval). We use it to calculate the average queue length  $E[q_i]$  and the packet rejection probability  $p_i$  which is then used to define new value of  $\lambda$ :  $\lambda_{i+1} = \lambda_i + \Delta\lambda_i$  due to the AQM algorithm, where

$$\Delta\lambda_i = \frac{\mu_i}{E[q_i]} - \frac{\lambda_i^2}{2} \frac{E[q_i]}{\mu} p_i. \quad (16)$$

The calculations are repeated over time ( $t_{i+1} = t_i + 1/\lambda_i$ ) for new values of  $\lambda$ . We assume that the  $i$ -th flow can start or end a transmission at any moment. The change in the source intensity  $\Delta\lambda$  at the time  $t$  affects the dispatching time and the queue length.

In a network model, the sum of arrivals at station  $j$  during time  $\Delta t$  has normal distribution with the average

$$\lambda_j \Delta t = \left[ \sum_{i=1}^M \lambda_i r_{ij} + \lambda_{0j} \right] \Delta t, \quad (17)$$

where  $r_{ij}$  is the routing probability between station  $i$  and  $j$ , and  $M$  is the number of interconnected stations,

and variance

$$\sigma_{A_j}^2 \Delta t = \left[ \sum_{i=1}^M C_{D_i}^2 \lambda_i r_{ij} + C_{0_j}^2 \lambda_{0_j} \right] \Delta t. \quad (18)$$

where  $C_{D_i}^2$  is the squared coefficient of variation of interdeparture times at station  $i$ .

These equations, written for every station  $i$ , make a system that allows us to determine  $\lambda_i$ , and the squared coefficient of variation of the interarrival, arrival times at any station  $j = 1, \dots, M$

$$C_{A_j}^2 = \frac{1}{\lambda_j} \sum_{i=1}^M r_{ij} \lambda_i [(C_{D_i}^2 - 1)r_{ij} + 1] + \frac{C_{0_j}^2 \lambda_{0_j}}{\lambda_j}, \quad (19)$$

hence the parameters for the diffusion model of any station. Parameters  $\lambda_{0_j}$ ,  $C_{0_j}^2$  refer to the streams originating at the station  $j$ .

## 5. NUMERICAL RESULTS

This section is divided into two parts. The first one (Section 5.1) compares results of Fluid Flow approximation and the proposed Diffusion Approximation model in the case of a simple network. In the second part (Section 5.2), a more complex network is considered.

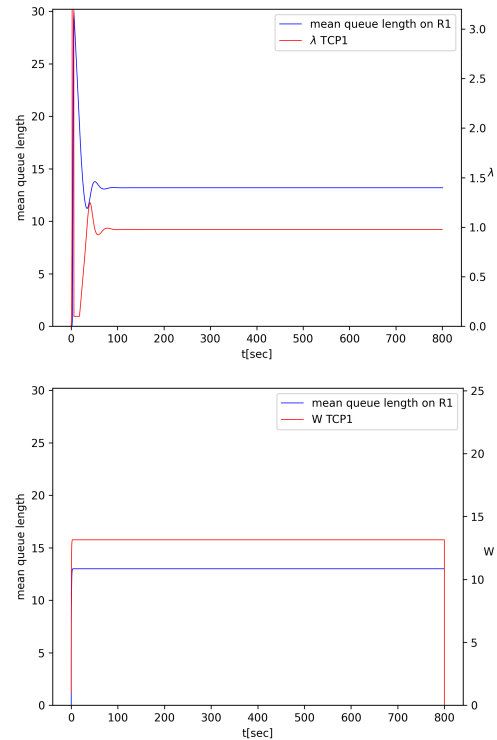
### 5.1. Comparison of the performance of the proposed model with the Fluid Flow approximation in a multi-device setup

The comparison of the developed Diffusion model with the Fluid Flow approximation was done for a network model consisting of 1, 2 or 3 routers and one transmitting and receiving station. We used three different AQM controllers: NLRED, RED and  $PI^\gamma$ . The following NLRED parameters were taken:  $a_1 = 0.00042$ ,  $a_2 = -0.0000038$ ,  $p_{\max} = 0.01$ ,  $w = 0.08$ . The RED parameters were:  $p_{\max} = 0.01$ ,  $w = 0.08$ . We considered  $Min_{th} = 10$ ,  $Max_{th} = 20$  and  $= 30$  packets RED and NLRED AQM buffers. The impact of RED parameters on the network traffic was described in [56]. Higher values of these parameters result in an increase in the network traffic fluctuations and their proper choice makes the control more transparent in the case of TCP/UDP flows, (see [22]). The values of parameters obtained in such a manner slightly differ from the ones proposed in the literature [57]. In the case of the NLRED algorithm, we refer to the values proposed in [11] as the ones offering the best transmission performance. We also used the following values for  $PI^\gamma$  controller: setpoint = 10,  $P = 0.0001$ ,  $I = 0.0004$ , buffer size (measured in packets) = 30, and  $\gamma = -0.1$ . Generally, it is difficult to choose the AQM/PI controller parameters. They strongly affect the packet dropping function, as integral order  $\gamma$  strengthens and accelerates the response of the controller.

Figures 1–31 present the queue behaviour for various AQM mechanisms, types and numbers of network flows present the queue (i.e. single TCP stream, single TCP and single UDP streams or one TCP and two UDP streams). Changes in the

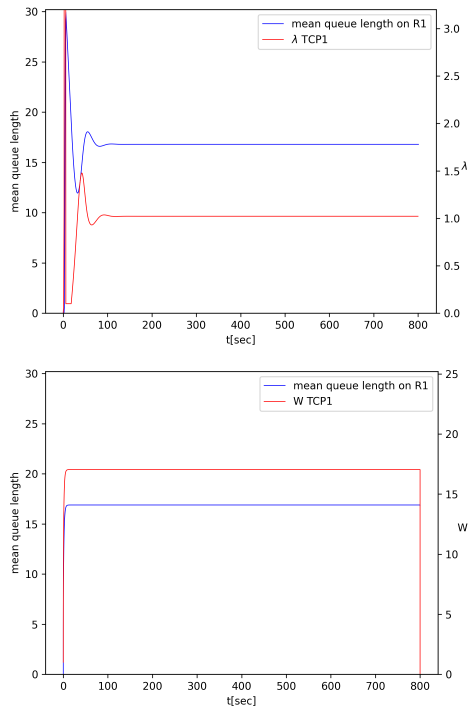
queue are determined by two mathematical models: Diffusion Approximation and Fluid Flow Approximation.

Figures 1–3 display the influence of NewReno TCP algorithm on one queue length in a network consisting of the single Router R1. The queue is controlled by three types of AQM mechanisms: NLRED (Fig. 1) RED (Fig. 2), and  $PI^\gamma$  (Fig. 3). As can be seen, the results obtained with both methods are similar. Average queue lengths are almost the same. However, the Diffusion model better reflects the transient queue behaviour. In steady-state, both models stabilise queue occupancy at the same level. Obtained average queue lengths depend on the AQM mechanism and take the following values: NLRED 13.0, RED 16.8 and  $PI^\gamma$  18.0.

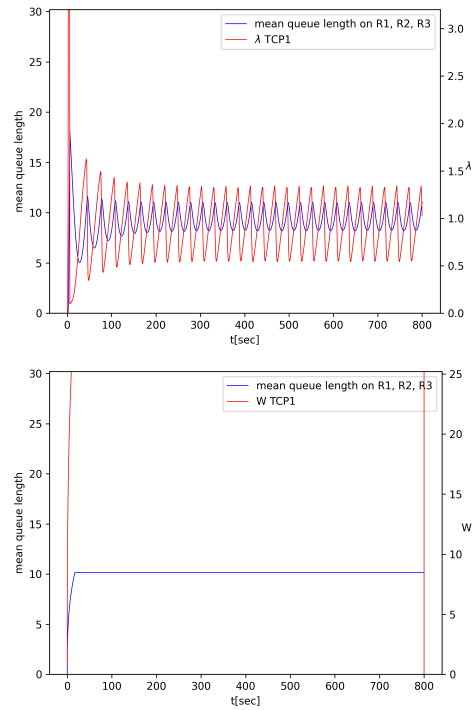


**Fig. 1.** The router mean queue length (blue) and throughput (red), 1 TCP stream and 1 router between sender and receiver for NLRED controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)

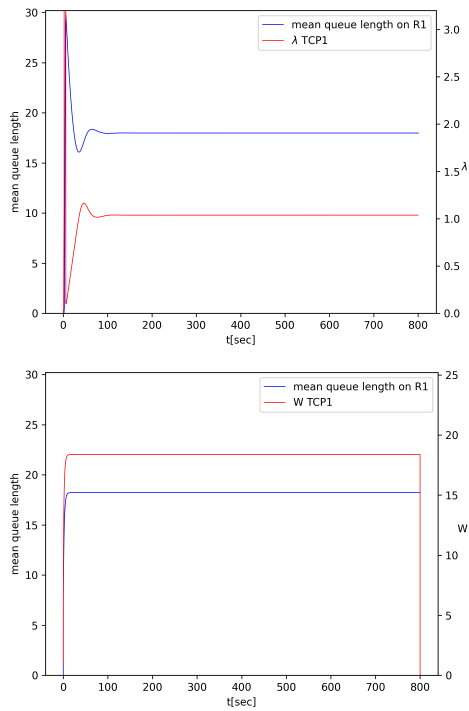
In the next stage of the experiments, we increase the number of considered routers. Figures 4–6 present results for a single TCP stream and network consisting of three routers: R1, R2 and R3. Similarly to the previous experiment, the figures present queues controlled by the following AQMs: NLRED (Fig. 4), RED (Fig. 5) and  $PI^\gamma$  (Fig. 6). The results show the advantage of diffusion in the case of transients and the consistency of results obtained by both methods. The following average queue lengths have been obtained: 10.2 for NLRED, 10.6 for RED and 10.8 for  $PI^\gamma$ . These values actually came close to the  $Min_{th}$  (RED and NLRED) and setpoint ( $PI^\gamma$ ) pa-



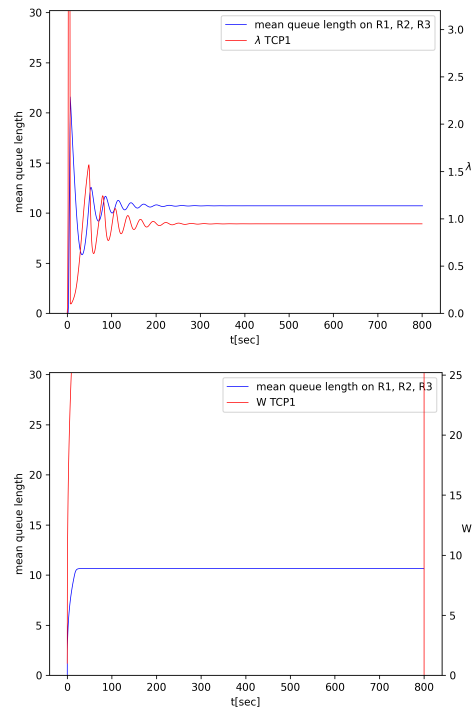
**Fig. 2.** The router mean queue length (blue) and throughput (red), 1 TCP stream and 1 router between sender and receiver for RED controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)



**Fig. 4.** The router mean queue length (blue) and throughput (red), 1 TCP stream and 3 routers between sender and receiver for NLRED controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)

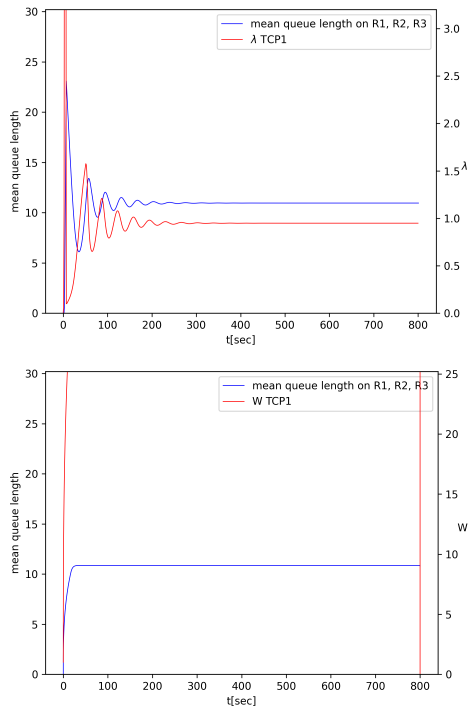


**Fig. 3.** The router mean queue length (blue) and throughput (red), 1 TCP stream and 1 router between sender and receiver for  $PI^\gamma$  controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)



**Fig. 5.** The router mean queue length (blue) and throughput (red), 1 TCP stream and 3 routers between sender and receiver for RED controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)



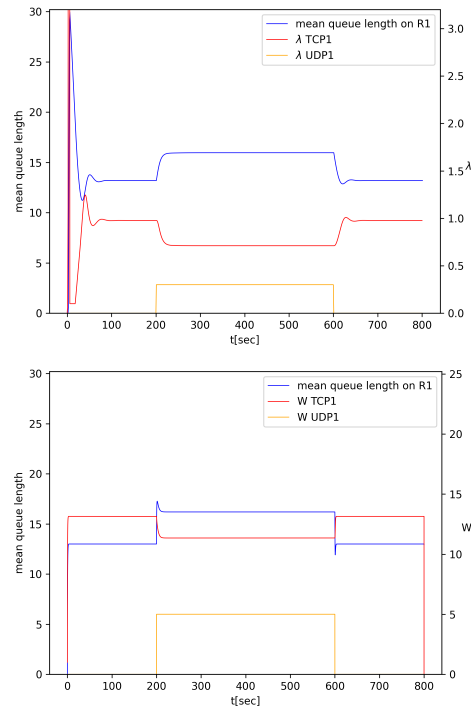


**Fig. 6.** The router mean queue length (blue) and throughput (red), 1 TCP stream and 3 routers between sender and receiver for  $PI^\gamma$  controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)

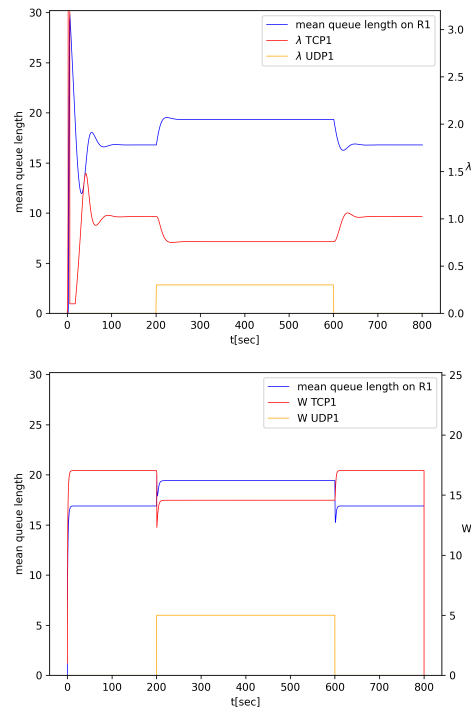
rameters. The decrease in queue occupancy is due to the reduction of the network load (a more significant number of nodes handles incoming traffic). An additional increase in the number of routers causes a further reduction in the length of the queues.

Figures 7–9 show the behaviour of the network with one router R1 in the case of one TCP and one UDP stream. The examined TCP stream starts at time  $t = 0$ . The UDP stream starts at time  $t = 200$  and finishes at  $t = 600$ . We have used a constant intensity of UDP stream. Its intensity is set to the level  $\lambda_{UDP1} = 0.3$  for the Diffusion Approximation model and  $W_{UDP1} = 5.0$  for the Fluid Flow model.

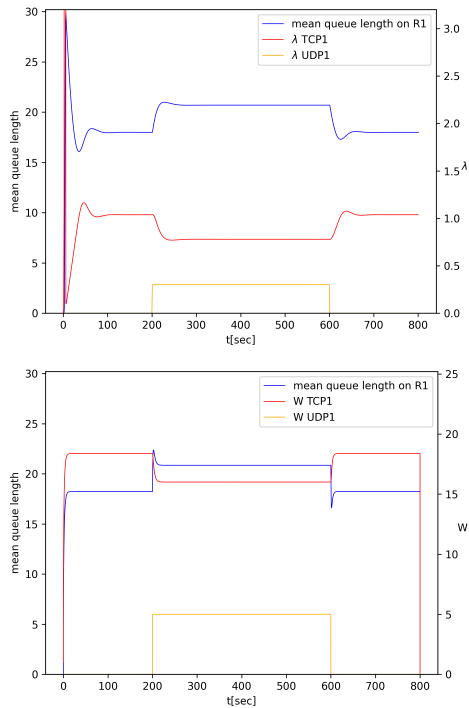
The results show that between  $t = 0$  and  $t = 200$ , the queue evolves similarly to the case of one TCP stream. The UDP stream (see  $t = 200$ ) decreases the TCP intensity and increases the queue occupancy. The queue length grows and stabilizes at the level 16.0 for NLRED (see Fig. 7), 19.3 for RED (Fig. 8) and 20.7 for  $PI^\gamma$  (Fig. 9). The experiment was then repeated with the number of routers increased to 3 (Figs. 10–12). The same as in the previous experiment, for 3 routers and one TCP source, the queue occupancy fluctuates with the minimum value of  $Min_{th}$  for RED and NLRED and setpoint = 10 for  $PI^\gamma$ . For all AQMs, the Diffusion Approximation better reflects the queue transients. This can be seen in the case of the NLRED queue. The captured oscillations are imperceptible when using the fluid flow method.



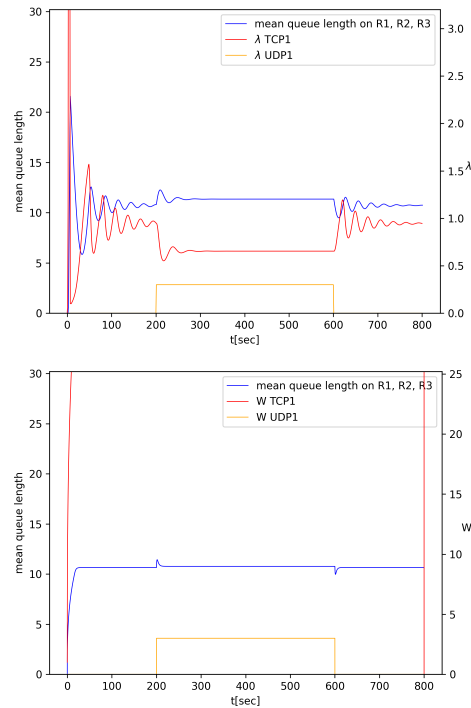
**Fig. 7.** The router mean queue length (blue) and throughput (red, yellow), 1 TCP and 1 UDP stream, 1 router between sender and receiver for NLRED controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)



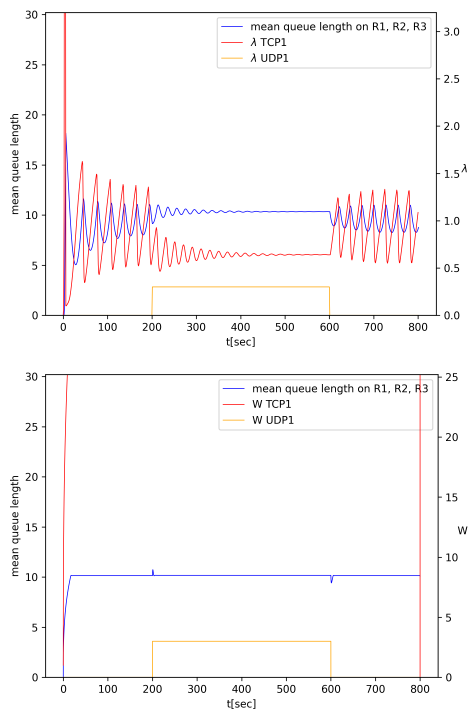
**Fig. 8.** The router mean queue length (blue) and throughput (red, yellow), 1 TCP and 1 UDP stream, 1 router between sender and receiver for RED controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)



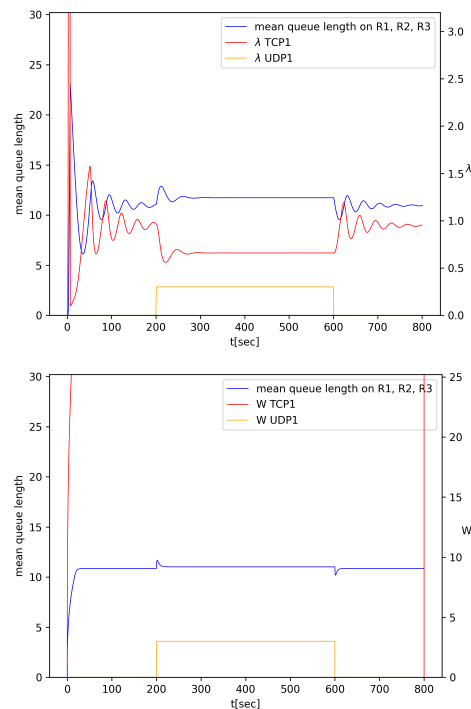
**Fig. 9.** The router mean queue length (blue) and throughput (red, yellow), 1 TCP and 1 UDP stream, 1 router between sender and receiver for  $PI^\gamma$  controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)



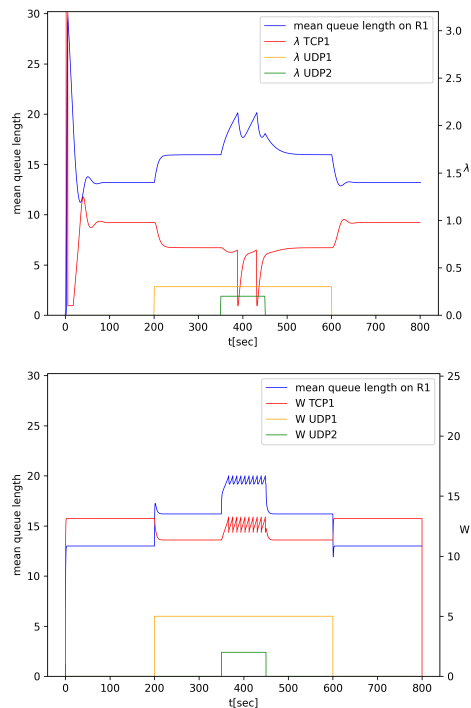
**Fig. 11.** The router mean queue length (blue) and throughput (red, yellow), 1 TCP and 1 UDP stream, 3 routers between sender and receiver for RED controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)



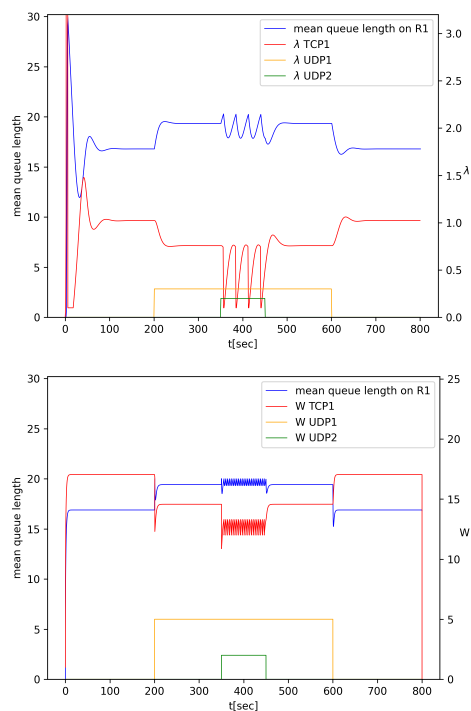
**Fig. 10.** The router mean queue length (blue) and throughput (red, yellow), 1 TCP and 1 UDP stream, 3 routers between sender and receiver for NLRED controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)



**Fig. 12.** The router mean queue length (blue) and throughput (red, yellow), 1 TCP and 1 UDP stream, 3 routers between sender and receiver for  $PI^\gamma$  controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)

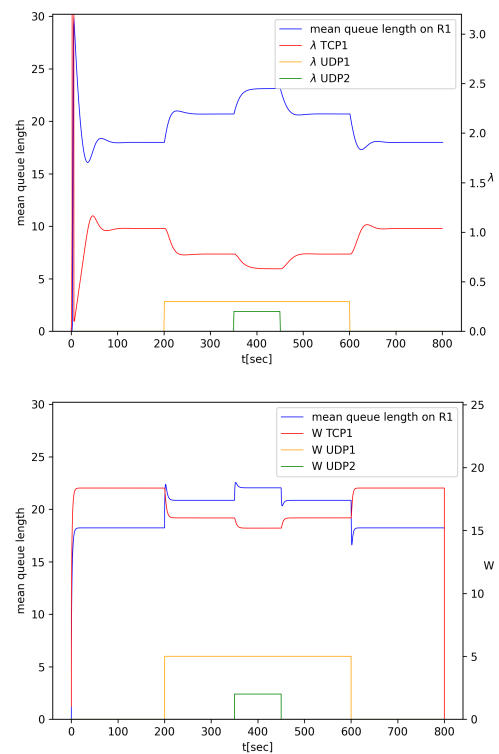


**Fig. 13.** The router mean queue length (blue) and throughput (red, yellow, green), 1 TCP and 2 UDP streams, 1 router between sender and receiver for NLRED controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)



**Fig. 14.** The router mean queue length (blue) and throughput (red, yellow, green), 1 TCP and 2 UDP streams, 1 router between sender and receiver for RED controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)

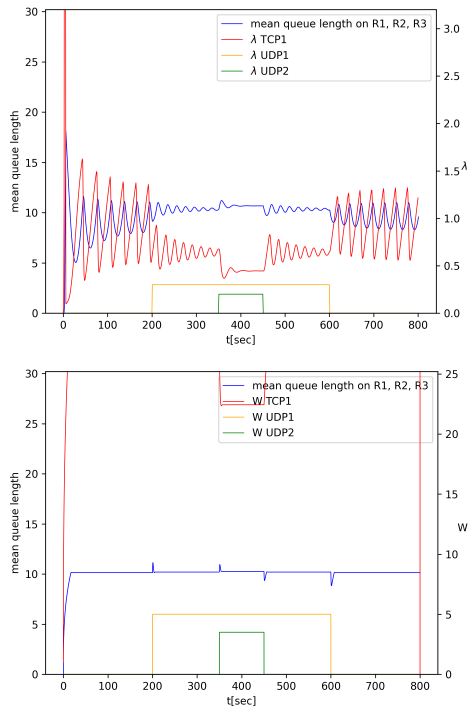
The influence of one TCP stream and two UDP streams on the queue occupancy for one router is presented in Figs. 13–15. They have used the following sequence of the experiments: at time  $t = 0$  TCP sender starts transmission, first UDP stream starts transmission at time  $t = 200$ , and the second one - at time  $t = 350$ , and finishes at  $t = 450$ ; at  $t = 600$  first UDP stream ends data sending. Both UDP streams reflect CBR (Constant Bit Rate) flows. The intensity of the streams was set to  $\lambda_{UDP1} = 0.3$  for Diffusion Approximation and  $W_{UDP1} = 5.0$  for Fluid Flow,  $\lambda_{UDP2} = 0.2$  for Diffusion Approximation and  $W_{UDP2} = 3.5$  for Fluid Flow. Each start or stop of the UDP stream affects the queue behaviour and the evolution of the congestion window of the TCP sender. The Diffusion model better captures these changes.



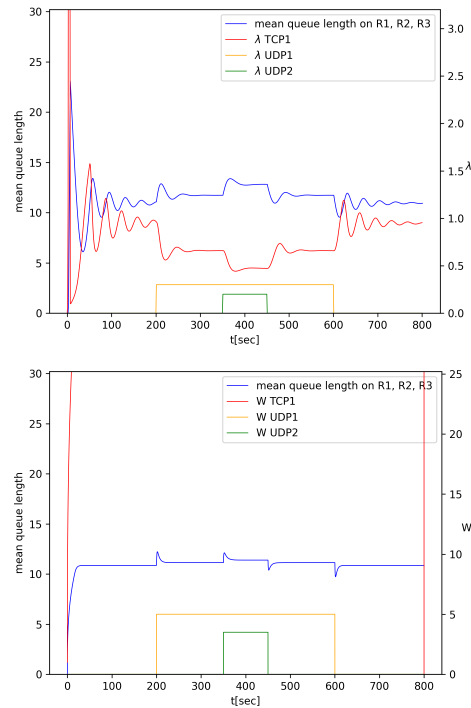
**Fig. 15.** The router mean queue length (blue) and throughput (red, yellow, green), 1 TCP and 2 UDP streams, 1 router between sender and receiver for  $PI^\gamma$  controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)

In the case of 3 routers (Figs. 16–18) (similar to previous experiments) the queue length oscillates around the assumed thresholds.

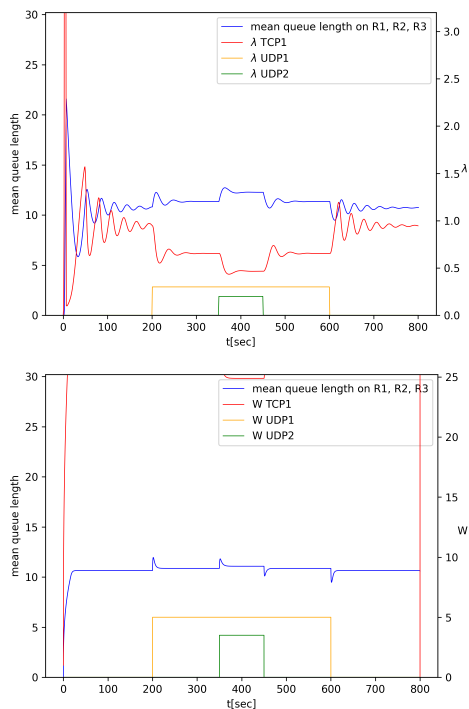
Figures 19–21 present the aggregate results of the average queue size for all experiments conducted with 1, 2, 3, and 4 routers. As it could be observed in previous experiments, within a single experiment, the queueing behaviour is identical for all routers. The figures also show that each additional router decreases the average occupancy of queues.



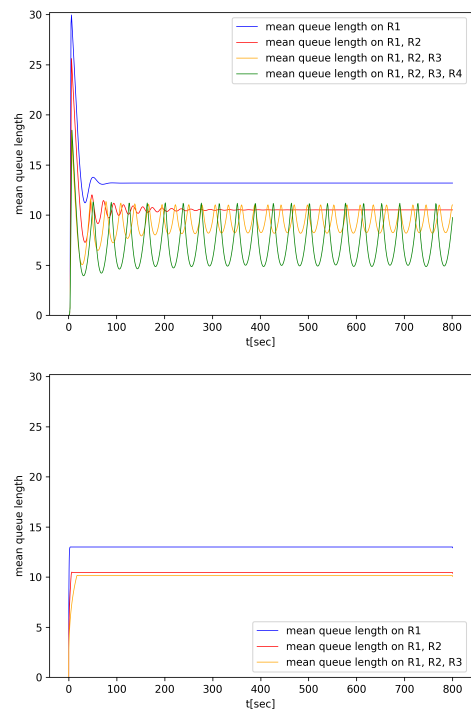
**Fig. 16.** The router mean queue length (blue) and throughput (red, yellow, green), 1 TCP and 2 UDP streams, 3 routers between sender and receiver for NLRED controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)



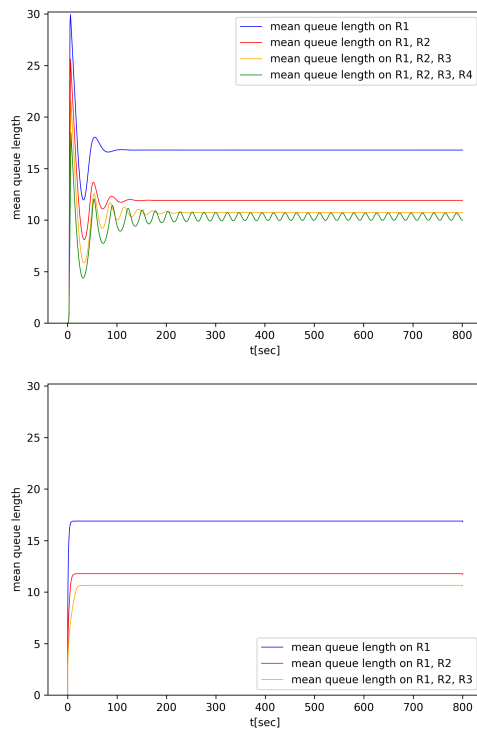
**Fig. 18.** The router mean queue length (blue) and throughput (red, yellow, green), 1 TCP and 2 UDP streams, 3 routers between sender and receiver for  $PI^\gamma$  controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)



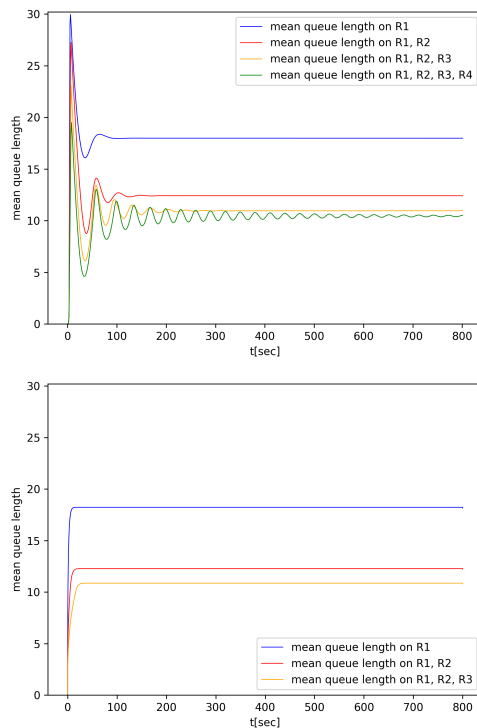
**Fig. 17.** The router mean queue length, 1 TCP and 2 UDP streams (blue) and throughput (red, yellow, green), 3 routers between sender and receiver for RED controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)



**Fig. 19.** The router mean queue length, 1 TCP stream for NLRED controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)



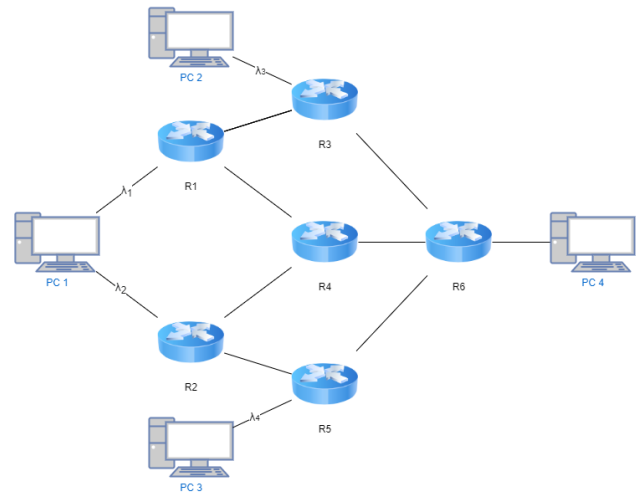
**Fig. 20.** The router mean queue length, 1 TCP stream for RED controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)



**Fig. 21.** The router mean queue length, 1 TCP stream for  $PI'$  controller, Diffusion Approximation (top), Fluid Flow approximation (bottom)

## 5.2. Diffusion Approximation for complex network topology

The study of the Diffusion model was conducted based on a network model consisting of 6 routers, three transmitting stations and one receiving station, Fig. 22.



**Fig. 22.** Diagram of a complex network model

**Router Details:** R1:  $\mu = 1, C_B^2 = 1$ , R2:  $\mu = 1, C_B^2 = 1$ , R3:  $\mu = 1.2, C_B^2 = 1$ , R4:  $\mu = 1.5, C_B^2 = 1$ , R5:  $\mu = 1.2, C_B^2 = 1$ , R6:  $\mu = 3, C_B^2 = 1$ .

The network structure is no longer linear. In addition, the Diffusion Approximation model allows us (in the case of multiple outputs from a single node) to set the percentage of traffic that will be redirected to a specified node.

### Verification of the correctness of the obtained results using simulation:

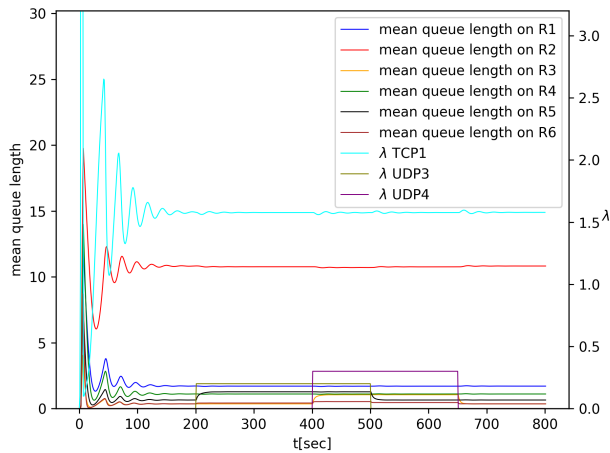
To verify the correctness of the diffusion approximation results, tests were carried out using simulation methods. A simulation model was created using SimPy and Python [58]. To show the realistic behavior of queues, the simulation was repeated 100,000 times. In Figs. 26–28, we present averaged results.

Figures 23–25 present the results obtained for one TCP stream and two UDP streams.

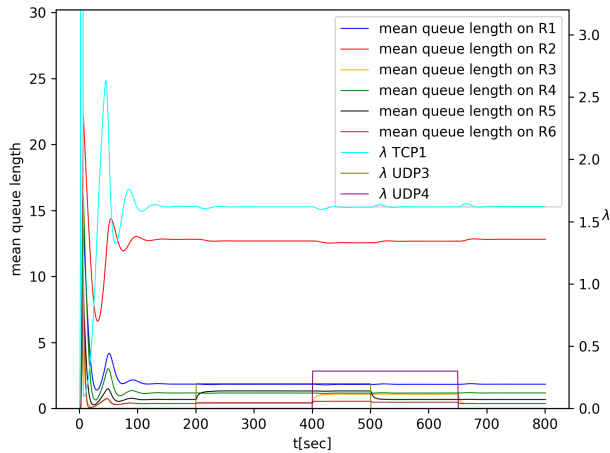
### Description of the experiment:

- The UDP3 stream enters router R5 ( $\lambda_4$ ) and begins at  $t = 200$  and ends at  $t = 500$  with a fixed intensity of 0.2.
- The UDP4 stream enters router R3 ( $\lambda_3$ ) and begins at  $t = 400$  and ends at  $t = 650$  with a constant intensity of 0.3.
- The TCP stream propagates to routers R1 ( $\lambda_1$ ) and R2 ( $\lambda_2$ ).

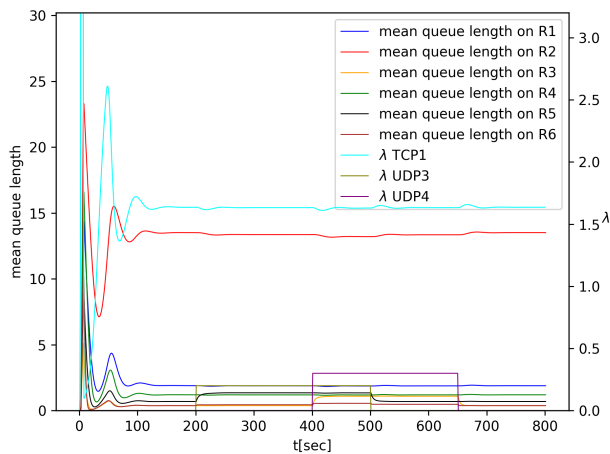
It can be seen that the incorporation of UDP streams hardly affects the behaviour of TCP streams. Therefore, the differences between different AQM mechanisms are barely noticeable. The obtained queue sizes were below the set thresholds. When a UDP3 stream (with constant intensity  $\lambda_{UDP3} = 0.3$ ) starts transmission at  $t = 200$  and ends at  $t = 500$ , the effect can be observed in the average occupancy of queue R5. In the case of



**Fig. 23.** The router mean queue length, 1 TCP and 2 UDP streams for NLRED controller



**Fig. 24.** The router mean queue length, 1 TCP and 2 UDP streams for RED controller



**Fig. 25.** The router mean queue length, 1 TCP and 2 UDP streams for  $PI^\gamma$  controller

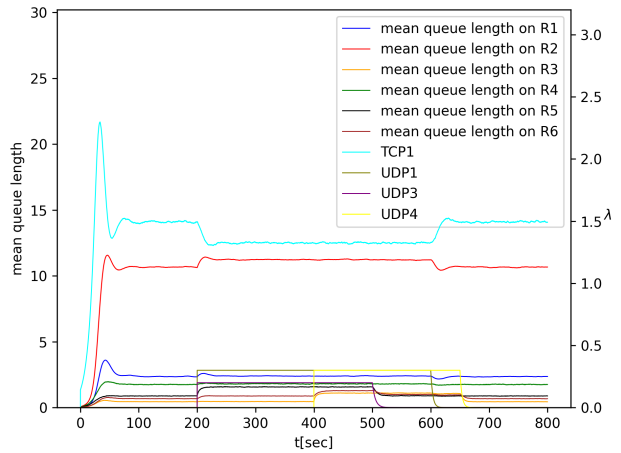
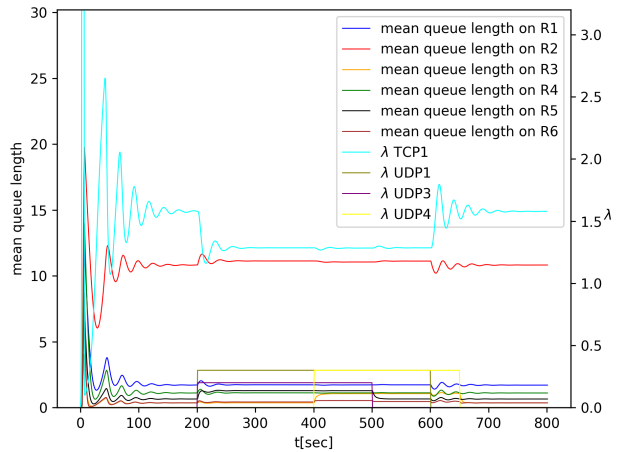
UDP4 stream (with constant intensity  $\lambda_{UDP4} = 0.3$ ), the average occupancy of queue R3 increases. Among all the controllers selected for the study, the highest average queue occupancy was observed for the  $PI^\gamma$  controller (see Fig. 25) and the lowest for the NLRED controller (see Fig. 23).

Figures 26–28 present the results for one TCP stream and three UDP streams.

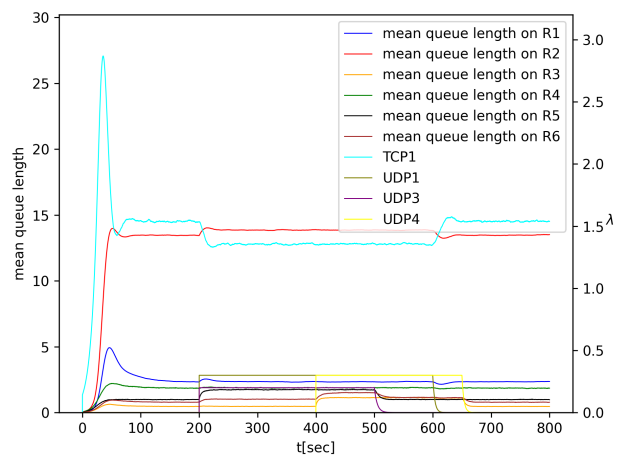
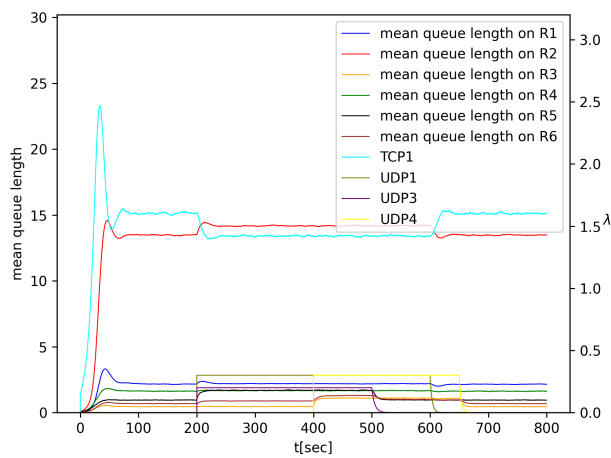
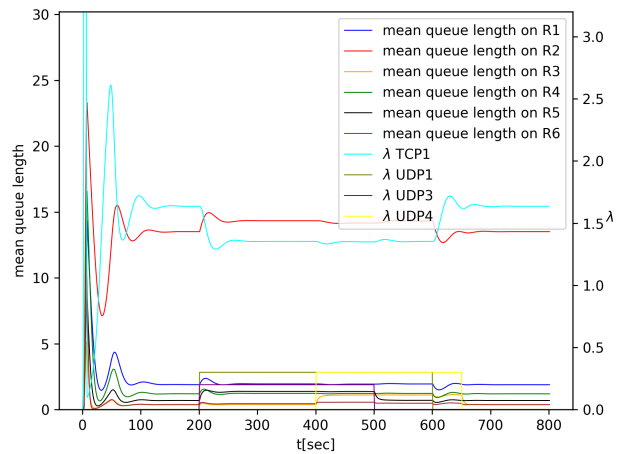
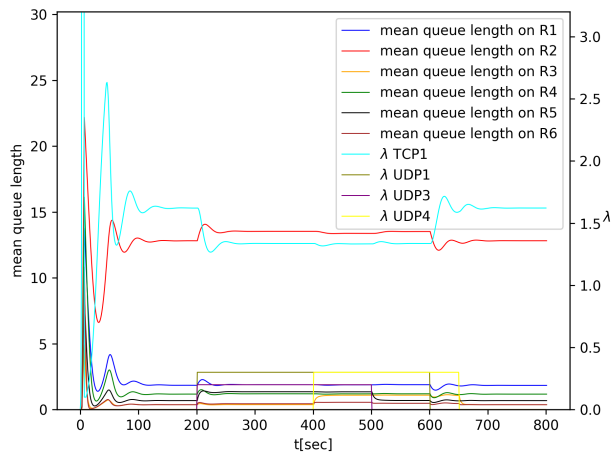
**Description of the experiment:**

- The UDP3 begins at  $t = 200$  and ends at  $t = 500$ ; it enters router R5 with fixed intensity  $\lambda_4 = 0.2$ .
- The UDP4 begins at  $t = 400$  and ends at  $t = 650$ ; it enters router R3 with fixed intensity  $\lambda_3 = 0.3$ .
- The UDP1 stream propagates through routers R1 ( $\lambda_1$ ) and R2 ( $\lambda_2$ ), begins at  $t = 200$  and ends at  $t = 600$  with a constant intensity of 0.3.
- The TCP stream propagates through routers R1 ( $\lambda_1$ ) and R2 ( $\lambda_2$ ).

Such an experiment allowed us to show the influence of UDP streams on the TCP stream. However, it is not significant. It is because streams come from different sources and spread in different parts of the network. At time  $t = 200$  an additional UDP1



**Fig. 26.** The router mean queue length, 1 TCP and 3 UDP streams for NLRED controller, Diffusion Approximation (top), Simulation (bottom)



**Fig. 27.** The router mean queue length, 1 TCP and 3 UDP streams for RED controller, Diffusion Approximation (top), Simulation (bottom)

**Fig. 28.** The router mean queue length, 1 TCP and 3 UDP streams for  $PI^\gamma$  controller, Diffusion Approximation (top), Simulation (bottom)

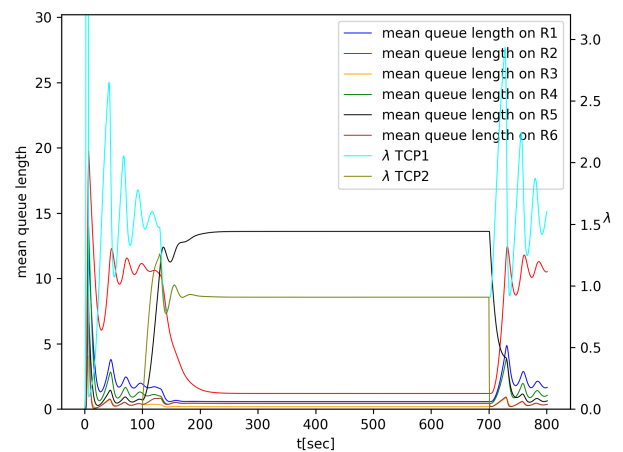
stream (of constant intensity  $\lambda_{UDP1} = 0.3$ ) begins the transmission. Until the end of the transmission at time  $t = 600$ , the average occupancy of queue R2 increases. The largest oscillation in queue occupancy during the start and stop of the additional stream transmission was observed for the NLRED controller (see Fig. 26). The most stable queue behavior and the highest average buffer occupancy values were observed for the  $PI^\gamma$  controller (see Fig. 28).

By comparing the values obtained with diffusion approximation (top part of Figs. 26–28) and simulation (see bottom part of Figs. 26–28), it should be noted that in the second case slightly higher average queue occupancy values are obtained. However, when the characteristics of the graphs are compared, the queueing behavior is similar.

Figures 29–31 present the obtained results for two TCP streams.

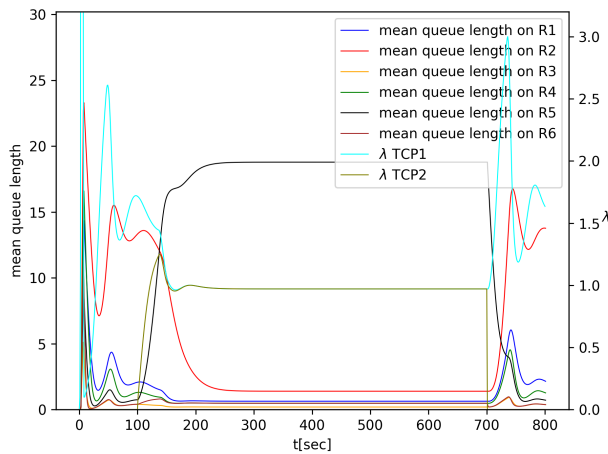
**Description of the experiment:**

- The TCP2 stream enters router R5 ( $\lambda_4$ ); it begins at  $t = 100$  and ends at  $t = 700$ .
- The TCP1 stream propagates to routers R1 ( $\lambda_1$ ) and R2 ( $\lambda_2$ ). Both streams seek to maximise the link utilisation. It affects

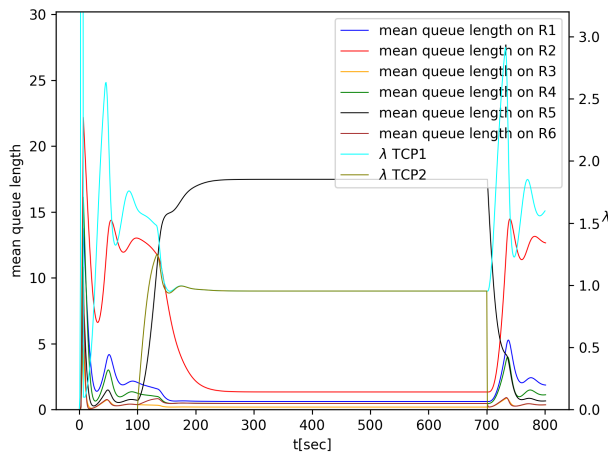


**Fig. 29.** The router mean queue length, 2 TCP streams for NLRED controller

mostly the nodes connected to the routing stations. The occupancy of the nodes which are far from the traffic sources is low.



**Fig. 30.** The router mean queue length, 2 TCP streams for RED controller



**Fig. 31.** The router mean queue length, 2 TCP streams for  $PI^\gamma$  controller

## 6. CONCLUSIONS

A good analysis of a computer network transient states, resulting from the changes of transmitted flows and the action of congestion control mechanisms, is essential in performance evaluation. In the article, we present a computer network model based on Diffusion Approximation, and we use it to evaluate the performance of TCP congestion window mechanisms cooperating with IP AQM mechanisms. The model allows multiple, existing simultaneously, TCP and UDP streams. We considered different types of AQM mechanisms (RED, NLRED,  $PI^\gamma$ ), determining packet loss probability in IP routers.

The proposed model was verified using the Fluid Flow approximation, which is traditionally used in such problems. Fluid Flow approximation operates on time-dependent mean values, and it is valid for Poisson traffic and exponential service times. Our experiments were limited to these assumptions and displayed that the results given by both methods are consistent,

which is a kind of validation of the diffusion model. However, the diffusion model allows us to assume any distribution of packets' interarrival times and service times; therefore, it is more general. Furthermore, diffusion models, based on means and variations of traffic fluctuations, allow us to capture queues dynamics (transient states) with more precision. It is also adapted to include routing probabilities, hence allowing us a better description of traffic distribution along the network and is closer to the actual operation of a computer network.

## ACKNOWLEDGEMENTS

This research was financed by National Science Center project no. 2017/27/B/ST6/00145.

## REFERENCES

- [1] J.M. Amigó, G. Duran, Á. Giménez, J. Valero, and O.M. Bonastre, "Modeling a new aqm model for internet chaotic behavior using petri nets," *Appl. Sci.*, vol. 11, no. 13, p. 5877, 2021.
- [2] N. El Fezazi, Y. Elfakir, F.A. Bender, and S. Idrissi, "Aqm congestion controller for tcp/ip networks: Multiclass traffic," *J. Control Autom. Electr. Syst.*, vol. 31, no. 4, pp. 948–958, 2020.
- [3] L. Tan, K. Huang, G. Peng, and G. Chen, "Stability of tcp/aqm networks under ddos attacks with design," *IEEE Trans. Network Sci. Eng.*, vol. 7, no. 4, pp. 3042–3056, 2020.
- [4] Wu-chang Feng, K.G. Shin, D.D. Kandlur, and D. Saha, "The BLUE Active Queue Management algorithms," *IEEE/ACM Trans. Networking*, vol. 10, no. 4, pp. 513–528, 2002.
- [5] Liujia Hu and A.D. Kshemkalyani, "HRED: a simple and efficient Active Queue Management algorithm," in *Proceedings. 13th International Conference on Computer Communications and Networks*, 2004, pp. 387–393.
- [6] C. Long, B. Zhao, X. Guan, and J. Yang, "The Yellow Active Queue Management algorithm," *Computer Networks*, vol. 47, no. 4, pp. 525–550, 2005.
- [7] N. Khademi, D. Ros, and M. Welzl, "The new aqm kids on the block: An experimental evaluation of codell and pie," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHOPS)*, 2014, pp. 85–90.
- [8] A. Adamu, V. Shorgin, S. Melnikov, and Y. Gaidamaka, "Flexible random early detection algorithm for queue management in routers," in *International Conference on Distributed Computer and Communication Networks*, 2020, pp. 196–208.
- [9] S.K. Bisoy and P.K. Pattnaik, "A neuron-based active queue management scheme for internet congestion control," *Int. J. Reasoning-based Intell. Syst.*, vol. 12, no. 4, pp. 238–247, 2020.
- [10] S. Floyd and V. Jacobson, "Random Early Detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 397–413, 1993, doi: [10.1109/90.251892](https://doi.org/10.1109/90.251892).
- [11] J. Domańska, D. Augustyn, and A. Domański, "The choice of optimal 3-rd order polynomial packet dropping function for NLRED in the presence of self-similar traffic," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 60, no. 4, pp. 779–786, 2012, doi: [10.2478/v10175-012-0090-x](https://doi.org/10.2478/v10175-012-0090-x).
- [12] K. Zhou, K. Yeung, and V. Li, "Nonlinear RED: A simple yet efficient Active Queue Management scheme," *Comput Networks, Int. J. Comput. Telecommun. Networking*, vol. 50, no. 18, pp. 3784–3794, Dec. 2006, doi: [10.1016/j.comnet.2006.04.007](https://doi.org/10.1016/j.comnet.2006.04.007).



- [13] I. Podlubny, "Fractional order systems and  $PI^\lambda D^\mu$  controllers," *IEEE Trans. Autom. Control*, vol. 44, no. 1, pp. 208–214, 1999.
- [14] M. Tarmizi, A. Albagul, O. Khalifa, and Wahyudi, "QoS Evaluation of Different TCPs Congestion Control Algorithm using NS2," in *2nd International Conference on Information Communication Technologies*, vol. 2, 2006, pp. 3222–3227.
- [15] S. Floyd, T. Henderson, and A. Gurtov, "The NewReno Modification to TCP's Fast Recovery Algorithm," *RFC 3782*, 2004.
- [16] C.A. Grazia, N. Patriciello, M. Klapez, and M. Casoni, "A cross-comparison between TCP and AQM algorithms: Which is the best couple for congestion control?" *14th International Conference on Telecommunications (ConTEL)*, pp. 75–82, 2017.
- [17] C.V. Hollot, V. Misra, and D. Towsley, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Transaction on Automatic Control*, vol. 47, no. 6, 2002.
- [18] P. Shah, S. Yasmin, S. Asghar, A. Qayyum, and H. Hasbullah, "A Fluid Flow Model for SCTP Traffic over the Internet," in *Proceedings of the International Conference on Emerging Technologies (ICET)*, 2012, pp. 1–6, doi: [10.1109/ICET.2012.6375481](https://doi.org/10.1109/ICET.2012.6375481).
- [19] J. Domańska, A. Domański, T. Czachórski, and J. Klamka, "Fluid flow approximation of time-limited TCP/UDP/XCP streams," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 62, no. 2, pp. 217–225, 2014, doi: [10.2478/bpasts-2014-0021](https://doi.org/10.2478/bpasts-2014-0021).
- [20] M. Bonaventura and R. Castro, "Fluid-flow and packet-level models of data networks unified under a modular/hierarchical framework: speedups and simplicity, combined," in *2018 Winter Simulation Conference (WSC)*, 2018, pp. 3825–3836.
- [21] O.J. Kravets, I.V. Atlasov, I.A. Aksenov, A.S. Molchan, O.Y. Frantsisko, and P.A. Rahman, "Increasing efficiency of routing in transient modes of computer network operation," *Int. J. Intell. Technol. Secur.*, vol. 13, no. 2, pp. 3–14, 2021.
- [22] A. Domański, J. Domańska, T. Czachórski, J. Klamka, J. Szygła, and D. Marek, "Diffusion Approximation Model of TCP NewReno Congestion Control Mechanism," *SN Comput. Sci.*, vol. 1, p. 43, 2020, doi: [10.1007/s42979-019-0032-x](https://doi.org/10.1007/s42979-019-0032-x).
- [23] T. Czachórski, "A method to solve Diffusion Equation with Instantaneous return Processes Acting as Boundary Conditions," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 41, no. 4, pp. 417–451, 1993.
- [24] E. Gelenbe and G. Pujolle, "The Behaviour of a Single-Queue in a General Queueing Network," *Acta Informatica*, vol. 7, no. 2, pp. 123–136, 1976, doi: [10.1007/BF00265766](https://doi.org/10.1007/BF00265766).
- [25] A. Duda, "Diffusion approximations for time-dependent queueing systems," *IEEE J. Sel. Areas Commun.*, vol. 4, pp. 905–918, 1986.
- [26] H. Seferoglu and E. Modiano, "TCP-aware backpressure routing and scheduling," in *Information Theory and Applications Workshop (ITA)*, 2014, pp. 1–9.
- [27] A. Bohloulzadeh and R. Mehri, "A Survey on Congestion Control Protocols in Wireless Sensor Networks," *Int. J. Wireless Inf. Networks*, vol. 27, pp. 365–384, 2020.
- [28] M.A. Jan, S. Jan, M. Alam, A. Akhuzada, and I. Rahman, "A Comprehensive Analysis of Congestion Control Protocols in Wireless Sensor Networks," *Mobile Networks Appl.*, vol. 23, pp. 456–468, 2018.
- [29] V. Misra, W. Gong, and D. Towsley, "Fluid-based analysis of network of AQM routers supporting TCP flows with an application to RED," *Computer Communication Review*, vol. 30, no. 4, pp. 151–160, 2000, doi: [10.1145/347059.347421](https://doi.org/10.1145/347059.347421).
- [30] W. Li, L. Zeng-zhi, C. Yan-ping, and X. Ke, "Fluid-based stability analysis of mixed TCP and UDP traffic under RED," in *10th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'05)*, 2005, pp. 341–348.
- [31] V. Misra, W.-B. Gong, and D. Towsley, "Fluid-Based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED," *Assoc. Comput. Mach.*, vol. 30, no. 4, p. 151–160, 2000, doi: [10.1145/347057.347421](https://doi.org/10.1145/347057.347421).
- [32] M. Nycz, T. Nycz, and T. Czachórski, "Modelling dynamics of TCP flows in very large network topologies," in *Information Sciences and Systems 2015*, 2016, pp. 251–259, doi: [10.1007/978-3-319-22635-4\\_23](https://doi.org/10.1007/978-3-319-22635-4_23).
- [33] Y. Hong and O.W.W. Yang, "Adaptive AQM controllers for IP routers with a heuristic monitor on TCP flows," *Int. J. Commun. Syst.*, vol. 19, no. 1, pp. 17–38, 2006.
- [34] Jinsheng Sun, K. Ko, Guanrong Chen, S. Chan, and M. Zukerman, "PD-RED: to improve the performance of RED," *IEEE Commun. Lett.*, vol. 7, no. 8, pp. 406–408, 2003.
- [35] F. Yanfie, R. Fengyuan, and L. Chuang, "Design a PID controller for Active Queue Management," in *Proc. of the Eighth IEEE Symposium on Computers and Communications. ISCC 2003*, vol. 2, 2003, pp. 985–990.
- [36] H. Unal, D. Melchor-Aguilar, D. Ustebay, S.-I. Niculescu, and H. Ozbay, "Comparison of PI controllers designed for the delay model of TCP/AQM," *Comput. Commun.*, vol. 36, no. 10, pp. 1225–1234, 2013.
- [37] G. Kahe and A.H. Jahangir, "A self-tuning controller for queuing delay regulation in TCP/AQM networks," *Telecommunication Systems*, vol. 71, pp. 215–229, 2019.
- [38] Y. Chen, I. Petras, and D. Xue, "Fractional order control – A tutorial," in *American Control Conference*, 2009, pp. 1397–1411.
- [39] W. Krajewski and U. Viaro, "On robust fractional order PI controller for TCP packet flow," in *BOS Conference: Systems and Operational Research*, Warsaw, Poland, Sep. 2014.
- [40] J. Domańska, A. Domański, T. Czachórski, and J. Klamka, "The use of a non-integer order PI controller with an Active Queue Management Mechanism," *Int. J. Appl. Math. Comput. Sci.*, vol. 26, pp. 777–789, 2016, doi: [10.1515/amcs-2016-0055](https://doi.org/10.1515/amcs-2016-0055).
- [41] J. Domańska, A. Domański, T. Czachórski, J. Klamka, and J. Szygła, "The AQM Dropping Packet Probability Function Based on Non-integer Order  $PI^\alpha D^\beta$  Controller," in *Lecture Notes in Electrical Engineering. Non-Integer Order Calculus and its Applications*, vol. 496, 2019, pp. 36–48, doi: [10.1007/978-3-319-78458-8\\_4](https://doi.org/10.1007/978-3-319-78458-8_4).
- [42] T. Czachórski and F. Pekergin, "Diffusion Approximation as a Modelling Tool," *Network Performance Engineering. A Handbook on Convergent Multi-Service Networks and Next Generation Internet. LNCS*, vol. 5233, pp. 447–476, 2011, doi: [10.1007/978-3-642-02742-0\\_20](https://doi.org/10.1007/978-3-642-02742-0_20).
- [43] T. Czachórski, K. Grochla, T. Nycz, and F. Pekergin, "A diffusion approximation model for wireless networks based on IEEE 802.11 standard," *Comput. Commun.*, vol. 33, pp. 86–92, 2010.
- [44] T. Nycz, M. Nycz, and T. Czachórski, "A Numerical Comparison of Diffusion and Fluid-Flow Approximations Used in Modelling Transient States of TCP/IP Networks," *Commun. Comput. Inf. Sci.*, vol. 431, pp. 213–222, 2014.
- [45] D. Marek, A. Domański, J. Domańska, T. Czachórski, J. Klamka, and J. Szygła, "Combined diffusion approximation – simulation model of AQM's transient behavior," *Comput. Commun.*, vol. 166, pp. 40–48, 2020, doi: [10.1016/j.comcom.2020.11.014](https://doi.org/10.1016/j.comcom.2020.11.014).
- [46] D. Marek, A. Domański, J. Domańska, J. Szygła, T. Czachórski, and J. Klamka, "Diffusion Model of a Non-Integer Order  $PI^\gamma$  Controller with TCP/UDP Streams," *Entropy*, vol. 23, no. 5, 2021, doi: [10.3390/e23050619](https://doi.org/10.3390/e23050619).

- [47] J. Domańska, A. Domański, T. Czachórski, and J. Klamka, "Self-similarity Traffic and AQM Mechanism Based on Non-integer Order  $PI^\alpha D^\beta$  Controller," in *Computer Networks: Communications in Computer and Information Science*, vol. 718, 2017, pp. 336–350, doi: [10.1007/978-3-319-59767-6\\_27](https://doi.org/10.1007/978-3-319-59767-6_27).
- [48] A. Domański, J. Domańska, T. Czachórski, J. Klamka, D. Marek, and J. Szyguła, "GPU Accelerated Non-integer Order  $PI^\alpha D^\beta$  Controller Used as AQM Mechanism," in *Computer Networks: Communications in Computer and Information Science*, vol. 860, 2018, pp. 286–299, doi: [10.1007/978-3-319-92459-5\\_23](https://doi.org/10.1007/978-3-319-92459-5_23).
- [49] I. Podlubny, *Fractional Differential Equations*. Academic Press, San Diego, USA, 1999, vol. 198.
- [50] M. Ciesielski and J. Leszczynski, "A Numerical Method for Solution of Ordinary Differential Equations of Fractional Order," in *Parallel Processing and Applied Mathematics*, vol. 2328, 2002, pp. 695–702, doi: [10.1007/3-540-48086-2\\_77](https://doi.org/10.1007/3-540-48086-2_77).
- [51] C. Hollot, V. Misra, and D. Towsley, "A control theoretic analysis of RED," in *Proc. IEEE INFOCOM 2001*, 2001, pp. 1510–1519.
- [52] D. Towsley, W. Gong, K. Hollot, Y. Liu, and V. Misra, "Fluid Methods for Modeling Large, Heterogeneous Networks," *NTIS*, 2005.
- [53] G.F. Newell, "Queues with time-dependent arrival rates. I – The transition through saturation," *J. Appl. Probab.*, vol. 2, no. 2, pp. 436–451, 1968, doi: [10.2307/3212264](https://doi.org/10.2307/3212264).
- [54] E. Gelenbe, "On Approximate Computer Systems Models," *J. ACM*, vol. 22, no. 2, pp. 261–269, 1975, doi: [10.1145/321879.321888](https://doi.org/10.1145/321879.321888).
- [55] E. Gelenbe, "A Diffusion Model for Packet Travel Time in a Random Multi-hop Medium," *ACM Trans. Sens. Netw.*, vol. 3, no. 2, p. 10, 2007, doi: [10.1145/1240226.1240230](https://doi.org/10.1145/1240226.1240230).
- [56] T. Bonald, M. May, and J. Bolot, "Analytic evaluation of RED performance," in *Proc. of INFOCOM*, 2000.
- [57] B. Zheng and M. Atiqzaman, "A Framework to Determine the Optimal Weight Parameter of RED in Next-Generation Internet Routers," *Int. J. Commun. Syst.*, vol. 21, no. 9, p. 987–1008, 2008, doi: [10.5555/1405579.1405584](https://doi.org/10.5555/1405579.1405584).
- [58] "Simpy documentation," [Accessed: 2022-05-27]. [Online]. Available: <https://simpy.readthedocs.io/en/latest/>

# Rozdział 11

## Podsumowanie

Głównym celem pracy była ocena wykorzystania koncepcji mechanizmu aktywnego zarządzania pakietami (AQM) w oparciu o kontroler  $PI^\alpha D^\beta$  niecałkowitych rzędów. Rezultaty eksperymentów badawczych zawartych w niniejszej rozprawie dowodzą postawionej w tej pracy tezie i pozwalają wysnuć następujące wnioski:

**Wniosek 1.** *Oparcie mechanizmu aktywnego zarządzania kolejką (AQM) o kontroler  $PI^\alpha D^\beta$  niecałkowitych rzędów pozwala na zmniejszenie średniej zajętości kolejki, a przez to na zwiększenie przepustowości transmisji danych w sieci Internet.*

**Wniosek 2.** *Modele aproksymacji dyfuzyjnej umożliwiają ocenę kooperacji strumieni TCP i UDP z mechanizmami aktywnego zarządzania kolejką (AQM). W porównaniu do najczęściej stosowanej metody aproksymacji Fluid-Flow, aproksymacja dyfuzyjna umożliwia pozyskanie bardziej szczegółowych wyników związanych z zachowaniem kolejki oraz pozwala na modelowanie bardziej złożonych struktur sieci rozległej.*

W niniejszej rozprawie wykorzystano symulator zdarzeń dyskretnych SimPy opracowany w języku Python oraz dwa modele analityczne - aproksymację Fluid-Flow oraz aproksymację dyfuzyjną. Modele analityczne zostały zaimplementowane w językach Python oraz C.

Uzyskane w ramach tej pracy wyniki służą ocenie zaproponowanych mechanizmów w porównaniu do już istniejących oraz powszechnie wykorzystywanych mechanizmów aktywnego zarządzania kolejką. Uzyskane wyniki wykazały, że zaproponowany w pracy mechanizm AQM bardzo dobrze nadaje się do regulowania

długości kolejki. W zależności od doboru parametrów kontrolera można uzyskać mechanizm AQM o różnych właściwościach. Regulując współczynniki elementu całkującego i proporcjonalnego kontrolera  $PI^\alpha D^\beta$  można zwiększać lub zmniejszać jego "moc". Kontroler "silniejszy" lepiej utrzymuje długość kolejki na poziomie oczekiwanym. Odbywa się to kosztem wzrostu liczby odrzuconych pakietów. Zmniejszając jego moc, zmniejszamy straty, kosztem wzrostu średniej długości kolejki. Badania wykazały również istotność rzędów całkowania. Na przykład zmniejszając jego wartość, zmniejszamy fluktuacje długości kolejki i pośrednio zmniejszamy zmienność natężenia ruchu sieciowego.

We wstępie pracy zaznaczono, że implementacja kontrolera niecałkowitych rzędów jest złożona obliczeniowo. Implementacje wykonane w strukturach GPU wykazały, że czas obliczeń odpowiedzi kontrolera  $PI^\alpha D^\beta$  nie będzie krytyczny, a przez to nie będzie powodował zmniejszenia prędkości transmisji.

Kierunek dalszych prac badawczych związany będzie ze stworzeniem dedykowanego oprogramowania umożliwiającego przeprowadzanie badań mechanizmu aktywnego zarządzania kolejką opartego o kontroler  $PI^\alpha D^\beta$  niecałkowitego rzędu w przełącznikach sieciowych. Celem tych prac będzie przeprowadzenie pełnych badań z wykorzystaniem rzeczywistej infrastruktury sieciowej. Rozwiązanie to oparte będzie o routery z systemem OpenWRT. W celu przyspieszenia czasu obliczeń kontrolera  $PI^\alpha D^\beta$  ponownie wykorzystane zostaną zweryfikowane już struktury GPU. Do jednej ze ścieżek związanej z tymi pracami należeć będzie również zaimplementowanie zaproponowanego modelu w macierzach programowalnych FPGA.

W ramach przyszłych prac rozwijany będzie również model aproksymacji dyfuzyjnej, który może być wykorzystany do analizy wydajności sieci definiowanej programowo SDN (ang. Software Defined Network). Pierwsze uzyskane wyniki zostały już przedstawione w artykule "Time-Dependent Performance of a Multi-Hop Software Defined Network" [75], którego jednak nie dołączono do listy publikacji zawartych w ramach niniejszej rozprawy.

# Bibliografia

- [1] Domański, A.: Wpływ mechanizmów protokołu TCP oraz algorytmów kolejowania na transmisję danych w sieci Internet. *Studia Informatica*, Vol. 38(1A), s. 3-245, 2017.
- [2] Floyd, S., Jacobson, V.: Random Early Detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, Vol. 1(4), 1993.
- [3] May, M., Diot, C., Lyles, B. and Bolot, J.: Influence of Active Queue Management parameters on aggregate traffic performance. Technical report, Institut de Recherche en Informatique et en Automatique, Rocquencourt, Le Chesnay, 2000.
- [4] Tan, L., Zhang, W., Peng, G. and Chen, G.: Stability of TCP/RED systems in AQM routers. *IEEE Transactions on Automatic Control*, Vol. 51(8), s. 1393 – 1398, 2006.
- [5] Floyd, S., Gummadi, R., Shenker, S.: Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management. *IEEE Transactions on Control Systems Technology*, 2011.
- [6] Zheng, B., Atiquzzaman, M.: DSRED: An Active Queue Management Scheme for New Generation Network, *Local Computer Networks*, 2000.
- [7] Zhou K., Yeung K. L., Li V.: Nonlinear RED: A simple yet efficient Active Queue Management scheme. *Elsevier Computer Networks*, Vol. 50, s. 3784 - 3794, 2006.
- [8] Domańska, J., Augustyn, D.R., Domański, A.: The choice of optimal 3-rd order polynomial packet dropping function for NLRED in the presence of self-similar traffic. *Bulletin of the Polish Academy of Sciences, Technical Sciences*, Vol. 60, 2012.

- [9] Floyd, S., Kohler, E.: Internet Research Needs Better Models. *Computer Communication Review*, Vol. 33(1), s. 29 - 34, 2003. doi:10.1145/774763.774767.
- [10] Domańska, J.: Markowowskie modele natężenia przesyłów internetowych. Instytut Informatyki Teoretycznej i Stosowanej Polskiej Akademii Nauk, 2014.
- [11] 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*, 2000. doi:10.1145/347059.347421.
- [12] Czachórski, T., Pekergin, F.: Diffusion Approximation as a Modelling Tool. *Network Performance Engineering. A Handbook on Convergent Multi-Service Networks and Next Generation Internet*. Springer Berlin Heidelberg, s. 447 - 476, 2011. doi:10.1007/978-3-642-02742-0\_20.
- [13] Domański, A., Domańska, J., Czachórski, T.: The Drop-From-Front Strategy in AQM. *Next Generation Teletraffic and Wired/Wireless Advanced Networking*, Springer Berlin Heidelberg, s. 61 - 72, 2007. doi:10.1007/978-3-540-74833-5\_6.
- [14] Sabatier, J., Agrawal, O.P., Tenreiro MacHado, J.A.: *Advances in fractional calculus: Theoretical developments and applications in physics and engineering*, 2007. doi:10.1007/978-1-4020-6042-7.
- [15] Li, H., Luo, Y., Chen, Y.: A Fractional Order Proportional and Derivative (FOPD) Motion Controller: Tuning Rule and Experiments. *IEEE Transactions on Control Systems Technology*, Vol. 18(2), s. 516 - 520, 2010. doi:10.1109/TCST.2009.2019120.
- [16] Sierociuk, D.: Estymacja i sterowanie dyskretnych układów dynamicznych ułamkowego rzędu opisanych w przestrzeni stanu. *Rozprawa Doktorska*, Politechnika Warszawska, Wydział Elektryczny, 2007.
- [17] Mittag-Leffler, G. M.: Sur la nouvelle fonction  $E_\alpha(x)$ . *C. R. Acad. Sci. Paris*, Vol. 137, s. 554 - 558, 1903.
- [18] Mittag-Leffler, G. M.: Spora la funzione  $E_\alpha(x)$ . *Rend. Acc. Lincei*, Vol. 5(13), s. 3 - 5, 1904.

- [19] Podlubny, I.: Fractional Differential Equations. Academic Press, San Diego, USA, Vol. 198, 1999.
- [20] Hilfer, R.: Application of Fractional Calculus in Physics. World Scientific, 2000. doi:10.1142/9789812817747\_fmatter.
- [21] Miller, K.S., Ross, B.: An Introduction to the Fractional Calculus and Fractional Differential Equations. Wiley, New York, 1993.
- [22] Oldham, K. B., Spanier, J.: The Fractional Calculus. Academic Press, 1974.
- [23] Caputo, M.: Linear Models of Dissipation whose Q is almost Frequency Independent — II. Geophysical Journal International, Vol. 13(5), s. 529 – 539, 1967. doi:10.1111/j.1365-246X.1967.tb02303.x.
- [24] Caputo, M.: Elasticità e dissipazione. Zanichelli, Bologna, 1969.
- [25] Ciesielski, M.: Frakcyjnalna metoda różnic skończonych w zastosowaniu do modelowania anomalnej dyfuzji w obszarze ograniczonym. Rozprawa doktorska, Politechnika Częstochowska, Wydział Inżynierii Mechanicznej i Informatyki, 2005.
- [26] Ostalczyk, P.: The non-integer difference of the discrete-time function and its application to the control system synthesis. International Journal of Systems Science, Vol. 31(12), s. 1551 – 1561, 2000. doi:10.1080/00207720050217322.
- [27] Hollot, C.V., Misra, V., Towsley, D., Gong, W-B.: On designing improved controllers for AQM routers supporting TCP flows. Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society, Vol. 3, s. 1726 - 1734, 2001. doi:10.1109/INFCOM.2001.916670.
- [28] Hollot, C.V., Misra, V., Towsley, D., Gong, W-B.: Analysis and design of controllers for AQM routers supporting TCP flows. IEEE Transactions on Automatic Control, Vol. 47(6), s. 945 - 959, 2002. doi:10.1109/TAC.2002.1008360.
- [29] Michiels, W., Melchor-Aquilar, D. and Niculescu, S.: Stability analysis of some classes of TCP/AQM networks. International Journal of Control, Vol. 79(9), s. 1136 – 1144, 2006. doi:10.1080/00207170600810701.

- [30] Melchor-Aquilar, D. and Castillo-Tores, V.: Stability analysis of proportional-integral AQM controllers supporting TCP flows. *Computacion y Sistemas*, Vol. 10(1), s. 401 – 414, 2007.
- [31] Ustebay, D., and Ozbay, H.: Switching resilient PI controllers for active queue management of TCP flows. *Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control*, London, UK, s. 574 - 578, 2007. doi:10.1109/ICNSC.2007.372842.
- [32] Melchor-Aquilar, D. and Niculescu, S.: Computing non-fragile PI controllers for delay models of TCP/AQM networks. *International Journal of Control*, Vol. 82(12), s. 2249 - 2259, 2009. doi:10.1080/00207170902984741.
- [33] Unal, H., Melchor-Aguilar, D., Ustebay, D., Niculescu, S.-I. and Ozbay, H.: Comparison of PI controllers designed for the delay model of TCP/AQM. *Computer Communications*, Vol. 36, s. 1225–1234, 2013. doi:10.1016/j.comcom.2013.03.001.
- [34] Podlubny, I.: Fractional order systems and  $PI^\lambda D^\mu$  controllers. *IEEE Transaction on Automatic Control*, Vol. 44, s. 208 - 214, 1999.
- [35] Chen, Y.Q., Petras, I., Xue, D.: Fractional order control - A tutorial. *American Control Conference*, s. 1397-1411, 2009.
- [36] Luo, Y., Chen, Y.Q.: Fractional-order proportional derivate controller for a class of fractional order systems. *Automatica*, Vol. 45(10), s. 2446-2450, 2009.
- [37] Oprzędkiewicz, K., Podsiadło, M.: The Fractional Order PID Control of the Forced Air Heating System. *Pomiary Automatyka Robotyka*, Vol. 23, s. 5-10, 2019. doi:10.14313/PAR\_231/5.
- [38] Bingi, K., Ibrahim, R., Karsiti, M., Hassan, S.: Frequency Response Based Curve Fitting Approximation of Fractional–Order PID Controllers. *International Journal of Applied Mathematics and Computer Science*, Vol. 29, s. 311 - 326, 2019. doi:10.2478/amcs-2019-0023.
- [39] Krajewski W., Viaro, U.: On robust fractional order PI controller for TCP packet flow. *BOS Conference: Systems and Operational Research*, Warsaw, Poland, s. 493 – 505, 2014.



- [40] Domańska, J., Domański, A., Czachórski, T., Klamka, J.: Self-similarity Traffic and AQM Mechanism Based on Non-integer Order  $PI^\alpha D^\beta$  Controller. Communications in Computer and Information Science. Springer International Publishing, Vol. 718, s. 336 - 350, 2017. doi:10.1007/978-3-319-59767-6\_27.
- [41] Domańska, J., Domański, A., Czachórski, T., Klamka, J.: The use of a non-integer order PI controller with an Active Queue Management Mechanism. International Journal of Applied Mathematics and Computer Science, Vol. 26, s. 777 - 789, 2016. doi:10.1515/amcs-2016-0055.
- [42] Domańska, J., Domański, A., Czachórski, T., Klamka, J., Szyguła, J.: The AQM Dropping Packet Probability Function Based on Non-integer Order  $PI^\alpha D^\beta$  Controller. Lecture Notes in Electrical Engineering, Springer International Publishing, Vol. 496, s. 36 - 48, 2019. doi:10.1007/978-3-319-78458-8\_4.
- [43] Domańska, J., Domański, A., Czachórski, T., Klamka, J., Marek, D., Szyguła, J.: GPU Accelerated Non-integer Order  $PI^\alpha D^\beta$  Controller Used as AQM Mechanism. Communications in Computer and Information Science, Springer Verlag, Vol. 860, s. 286 - 299, 2018. doi:10.1007/978-3-319-92459-5\_23.
- [44] Kaczorek, T.: Reachability and controllability to zero of positive fractional discrete-time systems. European Control Conference (ECC), s. 1708 - 1712, 2007.
- [45] Ciesielski, M., Leszczynski, J.: A Numerical Method for Solution of Ordinary Differential Equations of Fractional Order. Parallel Processing and Applied Mathematics. Springer Berlin Heidelberg, Vol. 2328, s. 695 - 702, 2002. doi:10.1007/3-540-48086-2\_77.
- [46] Domański, A., Domańska, J., Czachórski, T., Pagano, M.: The Fluid Flow Approximation of the TCP Vegas and Reno Congestion Control Mechanism. Computer and Information Sciences. Springer International Publishing, s. 193-200, 2016. doi:10.1007/978-3-319-47217-1\_21.
- [47] Gu, Y., Liu, Y., Towsley, D.: On integrating fluid model with packet simulation. IEEE INFOCOM, Vol. 4, s. 2856 - 2866, 2004. doi: 10.1109/INFOCOM.2004.1354702.

- [48] Kumar, S., Park, S.-J., Iyengar, S.-S.: A loss-event driven scalable fluid simulation method for high-speed networks. *International Journal of Computer and Telecommunications Networking*, Vol. 54 (1), s. 112 – 132, 2010. doi:10.1016/j.comnet.2009.08.018.
- [49] Hollot, C.V., Misra, V., Towsley, D., Gong W-B.: A control theoretic analysis of RED. *IEEE INFOCOM, Conference on Computer Communications*, Vol. 3, s. 1510 - 1519, 2001. doi:10.1109/INFCOM.2001.916647.
- [50] Towsley, D., Gong, W., Hollot, C.V., Liu, Y., Misra, V.: *Fluid Methods for Modeling Large, Heterogeneous Networks*, 2005.
- [51] Domańska, J., Domański A., Czachórski T., Klamka, J.: Fluid-Flow approximation of time-limited TCP/UDP/XCP streams. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, Vol. 62(2), s. 217-225, 2014. doi:10.2478/bpasts-2014-0021.
- [52] Nycz, M., Czachórski, T. Modelling of dynamics of TCP/IP transmissions with the RED mechanism. *Studia Informatica*, Vol. 33(3A), s. 49 - 62, 2012.
- [53] Nycz, T., Nycz, M., Czachórski, T.: A Numerical Comparison of Diffusion and Fluid-Flow Approximations Used in Modelling Transient States of TCP/IP Networks. *Communications in Computer and Information Science*, Springer, Cham, Vol. 431, s. 213 – 222, 2014. doi:10.1007/978-3-319-07941-7\_22.
- [54] Qi, Y., Ohtsuki, T.: QoE-Aware Video Streaming Transmission Optimization Method for Playout Threshold Adjustment in LTE. In *Proceedings of the 87th IEEE Vehicular Technology Conference (VTC Spring)*, s. 1 - 5, 3–6 June 2018. doi:10.1109/VTCSpring.2018.8417712.
- [55] Newell, G. F.: Queues with time-dependent arrival rates. I — The transition through saturation. *Journal of Applied Probability*, Vol. 2(2), s. 436 - 451, 1968. doi:10.2307/3212264.
- [56] Newell, G. F.: Queues with time-dependent arrival rates. II — The maximum queue and the return to equilibrium. *Journal of Applied Probability*, Vol. 2(3), s. 579 - 590, 1968. doi:10.2307/3211923.

- [57] Newell, G. F.: Queues with time-dependent arrival rates. III — A mild rush hour. *Journal of Applied Probability*, Vol. 2(3), s. 591 - 606, 1968. doi:10.2307/3211924.
- [58] Newell, G. F.: *Applications of Queueing Theory. Monographs on Statistics and Applied Probability*. Chapman and Hall, London, 1971. doi:10.1002/zamm.19730530310.
- [59] Kobayashi H.: Application of the Diffusion Approximation to Queueing Networks I: Equilibrium Queue Distributions. *Journal of the ACM (JACM)*, Vol. 21(2), s. 316 - 328, 1974. doi:10.1145/321812.321827.
- [60] Gelenbe, E.: On Approximate Computer Systems Models. *Journal of ACM*, Vol. 22(2), s. 261 - 269, 1975. doi:10.1145/321879.321888.
- [61] Iglehart, D., Whitt, W.: Multiple Channel Queues in Heavy Traffic - Part I. *Advances in Applied Probability*, Vol. 2(1), s. 150 - 177, 1970. doi:10.2307/3518347.
- [62] Iglehart, D., Whitt, W.: Multiple Channel Queues in Heavy Traffic. Part II: Sequences, Networks, and Batches. *Advances in Applied Probability*, Vol. 2(2), s. 355 - 369, 1970. doi:10.2307/1426324.
- [63] Whitt, W.: Multiple Channel Queues in Heavy Traffic - Part III: Random Server Selection. *Advances in Applied Probability*, Vol. 2(2), s. 370 - 375, 1970. doi:10.2307/1426325.
- [64] Iglehart, D.: Weak Convergence in Queueing Theory. *Advances in Applied Probability*, Vol. 5(3), s. 570 - 594, 1973. doi:10.2307/1425835.
- [65] Kobayashi H.: *Modelling and Analysis: An Introduction to System Performance Evaluation Methodology*. Addison-Wesley, Reading, Massachusetts, 1978.
- [66] Czachórski, T.: A method to solve Diffusion Equation with Instantaneous return Processes Acting as Boundary Conditions. *Bulletin of Polish Academy of Sciences, Technical Sciences*, Vol. 41(4), s. 417 - 451, 1993.
- [67] Cox, R. P., Miller, H. D.: *The Theory of Stochastic Processes*. Chapman and Hall, London, 1965.

- [68] Stehfest, H.: Algorithm 368: Numeric inversion of Laplace transform. Communications of the ACM, New York, USA, Vol. 13(1), s. 47 - 49, 1970. doi:10.1145/361953.361969.
- [69] Laalaoua, R, Atmaca, T., Jędrús, S., Czachórski, T.: Diffusion Model of RED Control Mechanism. Networking. Springer Berlin Heidelberg, s. 107 - 116, 2001. doi:10.1007/3-540-47728-4\_11.
- [70] Czachórski, T., Nycz, T., Pekergin, F.: Transient states analysis - diffusion approximation as an alternative to Markov models, fluid-flow approximation and simulation. IEEE Symposium on Computers and Communications, s. 13 - 18, 2009.
- [71] Czachórski, T.: A method to solve diffusion equation with instantaneous return processes acting as boundary conditions. Bull. Pol. Acad. Sci. Tech. Sci., Vol. 41, s. 417-451, 1993.
- [72] Czachórski, T., Grochla, K., Nycz, T., Pekergin, F.: Modeling the IEE 802.11 Networks MAC Layer Using Diffusion Approximation. Theoretical and Applied Informatics, Vol. 21 (3-4), s. 205-223, 2009.
- [73] Domański, A., Domańska, J., Czachórski, T., Klamka, J., Marek, D., Szyguła, J.: The Influence of the Traffic Self-similarity on the Choice of the Non-integer Order  $PI^\alpha$  Controller Parameters. Communications in Computer and Information Science. Springer International Publishing, Vol. 935, s. 76-83, 2018. doi:10.1007/978-3-030-00840-6\_9.
- [74] Domański, A., Domańska, J., Czachórski, T., Klamka, J., Szyguła, J., Marek, D.: AQM mechanism with the dropping packet function based on the answer of several  $PI^\alpha$  controllers. Communications in Computer and Information Science, Springer International Publishing, Vol. 1039, s. 400-412, 2019. doi:10.1007/978-3-030-21952-9\_29.
- [75] Czachórski, T.; Gelenbe, E.; Kuaban, G.S.; Marek, D: Time-Dependent Performance of a Multi-Hop Software Defined Network. Appl. Sci. 2021, vol. 11, 2469. doi:10.3390/app11062469.

## **Oświadczenia współautorstwa**

**Oświadczenia współautorów publikacji naukowych, określające indywidualny wkład każdego z nich w ich powstanie.**

**Declarations of co-authorship, detailing each co-author's individual contribution to each publication.**

dr hab. inż. Adam Domański, Prof. PŚ.  
Katedra Systemów Rozproszonych i Urządzeń Informatyki  
Politechnika Śląska  
ul. Akademicka 16, 44-100 Gliwice

Gliwice, 06.06.2022

### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Marek, D.; Szyguła, J. GPU accelerated non-integer order  $PI^\alpha D^\beta$  controller used as AQM mechanism. 25th International Conference on Computer Networks (CN 2018), Communications in Computer and Information Science, Springer International Publishing, vol. 860, pp. 286-299, 2018, [https://doi.org/10.1007/978-3-319-92459-5\\_23](https://doi.org/10.1007/978-3-319-92459-5_23)

mój udział [20 %] polegał na:

- współudziało w postawieniu problemu badawczego,
- współudziało w weryfikacji poprawności zastosowanego modelu matematycznego,
- współudziało w redakcji tekstu publikacji.

  
.....  
dr hab. inż. Adam Domański, Prof. PŚ.

dr hab. inż. Joanna Domańska  
Instytut Informatyki Teoretycznej i Stosowanej  
Polskiej Akademii Nauk  
ul. Bałtycka 5, 44-100 Gliwice

Gliwice, 14.06.2022

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Marek, D.; Szyguła, J. GPU accelerated non-integer order  $PI^\alpha D^\beta$  controller used as AQM mechanism. 25th International Conference on Computer Networks (CN 2018), Communications in Computer and Information Science, Springer International Publishing, vol. 860, s. 286-299, 2018. [https://doi.org/10.1007/978-3-319-92459-5\\_23](https://doi.org/10.1007/978-3-319-92459-5_23)

mój udział [5%] polegał na:

- współudziale w interpretacji uzyskanych wyników,
- współudziale w redakcji tekstu publikacji.



---

dr hab. inż. Joanna Domańska

Gliwice, 14.12.2021

Prof. dr hab. inż. Tadeusz Czachórski  
Instytut Informatyki Teoretycznej i Stosowanej  
Polskiej Akademii Nauk  
ul. Bałtycka 5, 44-100 Gliwice

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Marek, D.; Szyguła, J. GPU accelerated non-integer order PI $\alpha$ D $\beta$  controller used as AQM mechanism. 25th International Conference on Computer Networks (CN 2018), Communications in Computer and Information Science, Springer International Publishing, vol. 860, pp. 286-299, 2018, [https://doi.org/10.1007/978-3-319-92459-5\\_23](https://doi.org/10.1007/978-3-319-92459-5_23)

mój udział [ 5 ... %] polegał na:

- współudziale interpretacji uzyskanych wyników,
- współudziale w redakcji tekstu publikacji.



Prof. dr hab. inż. Tadeusz Czachórski



Gliwice, 14.12.2021

Prof. dr hab. inż. Jerzy Klamka  
Instytut Informatyki Teoretycznej i Stosowanej  
Polskiej Akademii Nauk  
ul. Bałtycka 5, 44-100 Gliwice

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Marek, D.; Szyguła, J. GPU accelerated non-integer order PI $\alpha$ D $\beta$  controller used as AQM mechanism. 25th International Conference on Computer Networks (CN 2018), Communications in Computer and Information Science, Springer International Publishing, vol. 860, pp. 286-299, 2018, [https://doi.org/10.1007/978-3-319-92459-5\\_23](https://doi.org/10.1007/978-3-319-92459-5_23)

mój udział [ 5,0 %] polegał na:

- współdziałanie w weryfikacji poprawności zastosowanego modelu matematycznego,
- współdziałanie w redakcji tekstu publikacji.



Prof. dr hab. inż. Jerzy Klamka

mgr inż. Jakub Szyguła  
Katedra Systemów Rozproszonych i Urządzeń Informatyki  
Politechnika Śląska  
ul. Akademicka 16, 44-100 Gliwice

Gliwice, 31.05.2022

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Marek, D.; Szyguła, J. GPU accelerated non-integer order  $PI^{\alpha}D^{\beta}$  controller used as AQM mechanism. 25th International Conference on Computer Networks (CN 2018), Communications in Computer and Information Science, Springer International Publishing, vol. 860, pp. 286-299, 2018, [https://doi.org/10.1007/978-3-319-92459-5\\_23](https://doi.org/10.1007/978-3-319-92459-5_23)

mój udział [5 %] polegał na:

- współudziale w analizie otrzymanych rezultatów,
- współudziale w redakcji tekstu publikacji.



mgr inż. Jakub Szyguła

dr hab. inż. Adam Domański, Prof. PŚ.  
Katedra Systemów Rozproszonych i Urządzeń Informatyki  
Politechnika Śląska  
ul. Akademicka 16, 44-100 Gliwice

Gliwice, 06.06.2022

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Marek, D.; Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Szyguła, J. Combined diffusion approximation–simulation model of AQM’s transient behavior. Computer Communications, vol. 166, pp. 40-48, 2021. <https://doi.org/10.1016/j.comcom.2020.11.014>

mój udział [15 %] polegał na:

- współudziało w postawieniu problemu badawczego,
- współudziało w interpretacji uzyskanych wyników,
- współudziało w redakcji i edycji tekstu publikacji.

  
.....  
dr hab. inż. Adam Domański, Prof. PŚ.

dr hab. inż. Joanna Domańska  
Instytut Informatyki Teoretycznej i Stosowanej  
Polskiej Akademii Nauk  
ul. Bałtycka 5, 44-100 Gliwice

Gliwice, 14.06.2022

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Marek, D.; Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Szyguła, J. Combined diffusion approximation–simulation model of AQM’s transient behavior. Computer Communications, vol. 166, pp. 40-48, 2021. <https://doi.org/10.1016/j.comcom.2020.11.014>

mój udział [5%] polegał na:

- współudziale w merytorycznej ocenie poprawności utworzonego modelu,
- współudziale w redakcji tekstu publikacji.



.....  
dr hab. inż. Joanna Domańska

Gliwice, 14.12.2021

Prof. dr hab. inż. Tadeusz Czachórski  
Instytut Informatyki Teoretycznej i Stosowanej  
Polskiej Akademii Nauk  
ul. Bałtycka 5, 44-100 Gliwice

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Marek, D.; Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Szyguła, J. Combined diffusion approximation–simulation model of AQM’s transient behavior. Computer Communications, vol. 166, pp. 40-48, 2021. <https://doi.org/10.1016/j.comcom.2020.11.014>

mój udział [ 5 %] polegał na:

- współudziałe w interpretacji uzyskanych wyników,
- współudziałe w redakcji i edycji tekstu publikacji.



Prof. dr hab. inż. Tadeusz Czachórski

Gliwice, 14.12.2021

Prof. dr hab. inż. Jerzy Klamka  
Instytut Informatyki Teoretycznej i Stosowanej  
Polskiej Akademii Nauk  
ul. Bałtycka 5, 44-100 Gliwice


#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Marek, D.; Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Szyguła, J. Combined diffusion approximation–simulation model of AQM’s transient behavior. Computer Communications, vol. 166, pp. 40-48, 2021. <https://doi.org/10.1016/j.comcom.2020.11.014>

mój udział [ 5,0 %] polegał na:

- współudziało w interpretacji uzyskanych wyników,
- współudziało w redakcji tekstu publikacji.

  
.....  
Prof. dr hab. inż. Jerzy Klamka

mgr inż. Jakub Szyguła  
Katedra Systemów Rozproszonych i Urządzeń Informatyki  
Politechnika Śląska  
ul. Akademicka 16, 44-100 Gliwice

Gliwice, 31.05.2022

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Marek, D.; Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Szyguła, J. Combined diffusion approximation–simulation model of AQM's transient behavior. Computer Communications, vol. 166, pp. 40-48, 2021. <https://doi.org/10.1016/j.comcom.2020.11.014>

mój udział [5 %] polegał na:

- współudziało w redakcji i edycji tekstu publikacji.



mgr inż. Jakub Szyguła

dr hab. inż. Adam Domański, Prof. PŚ.  
Katedra Systemów Rozproszonych i Urządzeń Informatyki  
Politechnika Śląska  
ul. Akademicka 16, 44-100 Gliwice

Gliwice, 06.06.2022


#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Szyguła, J.; Marek, D. Diffusion approximation model of TCP NewReno congestion control mechanism. Conference on Modelling Methods in Computer Systems, Networks and Bioinformatics, Paris. Springer Nature Computer Science, vol. 1 (43), 2019. <https://doi.org/10.1007/s42979-019-0032-x>

mój udział [20 %] polegał na:

- współudziałe w postawieniu problemu badawczego,
- współudziałe w weryfikacji poprawności zastosowanego modelu matematycznego,
- współudziałe w redakcji tekstu publikacji.

  
-----  
dr hab. inż. Adam Domański, Prof. PŚ.



dr hab. inż. Joanna Domańska  
Instytut Informatyki Teoretycznej i Stosowanej  
Polskiej Akademii Nauk  
ul. Bałtycka 5, 44-100 Gliwice

Gliwice, 14.06.2022

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Szyguła, J.; Marek, D. Diffusion approximation model of TCP NewReno congestion control mechanism. Conference on Modelling Methods in Computer Systems, Networks and Bioinformatics, Paris. Springer Nature Computer Science, vol. 1 (43), 2019. <https://doi.org/10.1007/s42979-019-0032-x>

mój udział [5%] polegał na:

- współudziale w merytorycznej ocenie poprawności utworzonego modelu,
- współudziale w redakcji tekstu publikacji.



dr hab. inż. Joanna Domańska

Gliwice, 14.12.2021

Prof. dr hab. inż. Tadeusz Czachórski  
Instytut Informatyki Teoretycznej i Stosowanej  
Polskiej Akademii Nauk  
ul. Bałtycka 5, 44-100 Gliwice

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Szyguła, J.; Marek, D. Diffusion approximation model of TCP NewReno congestion control mechanism. Conference on Modelling Methods in Computer Systems, Networks and Bioinformatics, 14-15 October 2019, Paris. Springer Nature Computer Science, vol. 1 (43), <https://doi.org/10.1007/s42979-019-0032-x>

mój udział [ 5 %] polegał na:

- współudziałe w weryfikacji poprawności zastosowanego modelu matematycznego,
- współudziałe w redakcji tekstu publikacji.



Prof. dr hab. inż. Tadeusz Czachórski

Gliwice, 14.12.2021

Prof. dr hab. inż. Jerzy Klamka  
Instytut Informatyki Teoretycznej i Stosowanej  
Polskiej Akademii Nauk  
ul. Bałtycka 5, 44-100 Gliwice


#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Szyguła, J.; Marek, D. Diffusion approximation model of TCP NewReno congestion control mechanism. Conference on Modelling Methods in Computer Systems, Networks and Bioinformatics, 14-15 October 2019, Paris. Springer Nature Computer Science, vol. 1 (43), <https://doi.org/10.1007/s42979-019-0032-x>

mój udział [ 5,0 %] polegał na:

- współudziałe w weryfikacji poprawności zastosowanego modelu matematycznego,
- współudziałe w redakcji tekstu publikacji.

  
.....  
Prof. dr hab. inż. Jerzy Klamka

mgr inż. Jakub Szyguła  
Katedra Systemów Rozproszonych i Urządzeń Informatyki  
Politechnika Śląska  
ul. Akademicka 16, 44-100 Gliwice

Gliwice, 31.05.2022

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Szyguła, J.; Marek, D. Diffusion approximation model of TCP NewReno congestion control mechanism. Conference on Modelling Methods in Computer Systems, Networks and Bioinformatics, Paris. Springer Nature Computer Science, vol. 1 (43), 2019. <https://doi.org/10.1007/s42979-019-0032-x>

mój udział [5 %] polegał na:

- współudziałe w dokonaniu przeglądu literatury,
- współudziałe w redakcji i edycji tekstu publikacji.



mgr inż. Jakub Szyguła

dr hab. inż. Adam Domański, Prof. PŚ.  
Katedra Systemów Rozproszonych i Urządzeń Informatyki  
Politechnika Śląska  
ul. Akademicka 16, 44-100 Gliwice

Gliwice, 06.06.2022

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Marek, D.; Domański, A.; Domańska, J.; Szyguła, J.; Czachórski, T.; Klamka, J. Diffusion Model of a Non-Integer Order PI $\lambda$  Controller with TCP/UDP Streams. Entropy, vol. 23(5):619, 2021.  
<https://doi.org/10.3390/e23050619>

mój udział [15 %] polegał na:

- współudziałe w postawieniu problemu badawczego, który pozwolił ocenić wpływ zastosowania kontrolera PI niecałkowitego rzędu jako mechanizmu AQM,
- współudziałe w weryfikacji poprawności zastosowanego modelu matematycznego,
- współudziałe w redakcji i edycji tekstu publikacji.



dr hab. inż. Adam Domański, Prof. PŚ.

dr hab. inż. Joanna Domańska  
Instytut Informatyki Teoretycznej i Stosowanej  
Polskiej Akademii Nauk  
ul. Bałtycka 5, 44-100 Gliwice

Gliwice, 14.06.2022

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Marek, D.; Domański, A.; Domańska, J.; Szyguda, J.; Czachórski, T.; Klamka, J. Diffusion Model of a Non-Integer Order Ply Controller with TCP/UDP Streams. Entropy, vol. 23(5):619, 2021.  
<https://doi.org/10.3390/e23050619>

mój udział [5%] polegał na:

- współudziale w określeniu problemu badawczego,
- współudziale w merytorycznej ocenie poprawności utworzonego modelu,
- współudziale w redakcji tekstu publikacji.



dr hab. inż. Joanna Domańska

mgr inż. Jakub Szyguła  
Katedra Systemów Rozproszonych i Urządzeń Informatyki  
Politechnika Śląska  
ul. Akademicka 16, 44-100 Gliwice

Gliwice, 31.05.2022

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Marek, D.; Domański, A.; Domańska, J.; Szyguła, J.; Czachórski, T.; Klamka, J. Diffusion Model of a Non-Integer Order PI $\lambda$  Controller with TCP/UDP Streams. Entropy, vol. 23(5):619, 2021. <https://doi.org/10.3390/e23050619>

mój udział [5 %] polegał na:

- współudziałe w zaplanowaniu i przeprowadzaniu badań zaproponowanego modelu,
- współudziałe w redakcji i edycji tekstu publikacji.



mgr inż. Jakub Szyguła

Gliwice, 14.12.2021

Prof. dr hab. inż. Tadeusz Czachórski  
Instytut Informatyki Teoretycznej i Stosowanej  
Polskiej Akademii Nauk  
ul. Bałtycka 5, 44-100 Gliwice

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Marek, D.; Domański, A.; Domańska, J.; Szyguła, J.; Czachórski, T.; Klamka, J. Diffusion Model of a Non-Integer Order Ply Controller with TCP/UDP Streams. Entropy, vol. 23(5):619, 2021. <https://doi.org/10.3390/e23050619>

mój udział [ 5 %] polegał na:

- współudziale w weryfikacji poprawności zastosowanego modelu matematycznego,
- współudziale w redakcji i edycji tekstu publikacji.



Prof. dr hab. inż. Tadeusz Czachórski



Gliwice, 14.12.2021

Prof. dr hab. inż. Jerzy Klamka  
Instytut Informatyki Teoretycznej i Stosowanej  
Polskiej Akademii Nauk  
ul. Bałtycka 5, 44-100 Gliwice


#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Marek, D.; Domański, A.; Domańska, J.; Szyguła, J.; Czachórski, T.; Klamka, J. Diffusion Model of a Non-Integer Order PI Controller with TCP/UDP Streams. Entropy, vol. 23(5):619, 2021.  
<https://doi.org/10.3390/e23050619>

mój udział [ 50 %] polegał na:

- współudziałe w interpretacji uzyskanych wyników,
- współudziałe w redakcji tekstu publikacji.

  
.....  
Prof. dr hab. inż. Jerzy Klamka

dr hab. inż. Adam Domański, Prof. PŚ.  
Katedra Systemów Rozproszonych i Urządzeń Informatyki  
Politechnika Śląska  
ul. Akademicka 16, 44-100 Gliwice

Gliwice, 05.08.2022

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Marek, D.; Domański, A.; Domańska, J.; Szyguła, J.; Czachórski, T.; Klamka, J.; Filus, K.  
Approximation Models for the Evaluation of TCP/AQM Networks. Bulletin of the Polish  
Academy of Sciences Technical Sciences, vol. 70(4), 2022. DOI: 10.24425/bpasts.2022.141986

mój udział [15 %] polegał na:

- współudziałe w postawieniu problemu badawczego,
- współudziałe w weryfikacji poprawności zastosowanego modelu matematycznego,
- współudziałe w redakcji tekstu publikacji.

  
.....  
dr hab. inż. Adam Domański, Prof. PŚ.

dr hab. inż. Joanna Domańska  
Instytut Informatyki Teoretycznej i Stosowanej  
Polskiej Akademii Nauk  
ul. Bałtycka 5, 44-100 Gliwice

Gliwice, 12.07.2022

### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Marek, D.; Domański, A.; Domańska, J.; Szyguła, J.; Czachórski, T.; Klamka, J.; Filus, K.  
Approximation Models for the Evaluation of TCP/AQM Networks. Bulletin of the Polish  
Academy of Sciences: Technical Sciences (BPASTS), vol. 70, 2022.

mój udział [5%] polegał na:

- współudziale w określeniu problemu badawczego,
- współudziale w redakcji tekstu publikacji.



dr hab. inż. Joanna Domańska

mgr inż. Jakub Szyguła  
Katedra Systemów Rozproszonych i Urządzeń Informatyki  
Politechnika Śląska  
ul. Akademicka 16, 44-100 Gliwice

Gliwice, 01.08.2022

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Marek, D.; Domański, A.; Domańska, J.; Szyguła, J.; Czachórski, T.; Klamka, J.; Filus, K.  
Approximation Models for the Evaluation of TCP/AQM Networks. Bulletin of the Polish  
Academy of Sciences Technical Sciences, vol. 70(4), 2022. DOI: 10.24425/bpasts.2022.141986

mój udział [5 %] polegał na:

- współudziało w analizie otrzymanych rezultatów,
- współudziało w redakcji tekstu publikacji.



mgr inż. Jakub Szyguła

Prof. dr hab. inż. Tadeusz Czachórski  
Instytut Informatyki Teoretycznej i Stosowanej  
Polskiej Akademii Nauk  
ul. Bałtycka 5, 44-100 Gliwice

Gliwice, 28.07.2022

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Marek, D.; Domański, A.; Domańska, J.; Szyguła, J.; Czachórski, T.; Klamka, J.; Filus, K.  
Approximation Models for the Evaluation of TCP/AQM Networks. Bulletin of the Polish  
Academy of Sciences: Technical Sciences, vol. 70, 2022.

mój udział [ 5 %] polegał na:

- współudziale w interpretacji uzyskanych wyników,
- współudziale w redakcji i edycji tekstu publikacji.



Prof. dr hab. inż. Tadeusz Czachórski

Prof. dr hab. inż. Jerzy Klamka  
Instytut Informatyki Teoretycznej i Stosowanej  
Polskiej Akademii Nauk  
ul. Bałtycka 5, 44-100 Gliwice

Gliwice, 28.07.2022

#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Marek, D.; Domański, A.; Domańska, J.; Szyguła, J.; Czachórski, T.; Klamka, J.; Filus, K.  
Approximation Models for the Evaluation of TCP/AQM Networks. Bulletin of the Polish  
Academy of Sciences: Technical Sciences, vol. 70, 2022.

mój udział [ 5,0 %] polegał na:

- współudziałe w weryfikacji poprawności zastosowanego modelu matematycznego,
- współudziałe w redakcji tekstu publikacji.



Prof. dr hab. inż. Jerzy Klamka

mgr inż. Katarzyna Filus  
Instytut Informatyki Teoretycznej i Stosowanej  
Polskiej Akademii Nauk  
ul. Bałtycka 5, 44-100 Gliwice

Gliwice, 13.07.2022

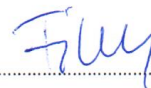
#### Oświadczenie współautorstwa

Oświadczam, że w pracy:

Marek, D.; Domański, A.; Domańska, J.; Szyguła, J.; Czachórski, T.; Klamka, J.; Filus, K.  
Approximation Models for the Evaluation of TCP/AQM Networks. Bulletin of the Polish  
Academy of Sciences: Technical Sciences (BPASTS), vol. 70, 2022.

mój wkład [5 %] polegał na:

- współudziało w redakcji i edycji tekstu publikacji.



mgr inż. Katarzyna Filus

**Informacja o otrzymanych  
nagrodach w czasie realizacji prac  
nad przygotowaniem rozprawy  
doktorskiej**





Politechnika  
Śląska



UCZELNIA  
BADAWCZA  
INICJATYWA DOSKONALOŚCI

**mgr inż.**  
**Dariusz Marek**

otrzymuje  
zespołową  
NAGRODĘ REKTORA  
stopnia III  
za osiągnięcia naukowe



  
prof. dr hab. inż.  
**Arkadiusz Mężyk**  
Rektor Politechniki Śląskiej

Gliwice, 01.10.2021 r.



# Certificate of Award

*This is to certify that*

**MISCEA.PL ENGINEERING SP. Z O.O.**

**MISCEA.PL'S TEAM – LESZEK REMIORZ, ERYK REMIORZ, OLEG ANTEMIJCZUK,  
ADRIAN CZAJKOWSKI, JAROSŁAW PADUCH, GABRIEL DRABIK, JAKUB SZYGUŁA,  
DARIUSZ MAREK, SEBASTIAN PAWLAK, MARCIN PASZKUTA, GRZEGORZ BARON**

*has been awarded the*

## **GOLD MEDAL**

*for the invention*

**A REVOLUTIONARY BREAKTHROUGH SHOWER SOLUTION**

*on the occasion of*



**2020 Kaohsiung International Invention & Design EXPO**

**10~12 December, 2020**

**Kaohsiung, Taiwan**

**Manli Hsieh**  
President of WIIPA

**Neven Marković**  
President of the International Jury



**World Invention Intellectual Property Associations**



2020 Hong Kong International Invention and Design Competition

# Silver Medal

For

A REVOLUTIONARY BREAKTHROUGH SHOWER SOLUTION

**Leszek Remiorz, Eryk Remiorz, Oleg Antemijczuk, Adrian Czajkowski,**

**Jarosław Paduch, Gabriel Drabik, Jakub Szygula,**

**Dariusz Marek, Sebastian Pawlak, Marcin Paszkuta, Grzegorz Baron**

*Jung Chuan Chou*

Jung-Chuan Chou  
Chairman of Jury member

*K.C. Wu*

Kou-Chen Wu  
Chairman of IIDC

4 Dec 2020

No.202040