Załącznik Z1:
Autoreferat przedstawiający opis dorobku naukowego oraz zasadniczego osiągnięcia habilitacyjnego pod tytułem:

Opracowanie podstaw teoretycznych syntezy wybranych mechanizmów sterowania dla systemów teleinformatycznych

W niniejszym referacie podsumowuję swój dorobek naukowy oraz działalność projektową liczone od 2010 roku, czyli od momentu uzyskania z wyróżnieniem stopnia doktora nauk technicznych. Przedstawiam swoje zasadnicze osiągnięcie naukowe, które stanowi podstawę do ubiegania się o stopień doktora habilitowanego w dziedzinie nauk technicznych w dyscyplinie informatyka, a także wyniki prac badawczo-rozwojowych, które w tym okresie uzyskałem. Swój dorobek podzieliłem na dwie części, teoretyczną i praktyczną. Podkreślam tym fakt, że w swojej pracy miałem okazję nie tylko formułować twierdzenia matematyczne opisujące interesujące mnie zjawiska, lecz także eksperymentować z ich praktycznym zastosowaniem. Eksperymenty te pozwoliły pozytywnie zweryfikować opracowane przeze mnie twierdzenia i zaowocowały stworzeniem projektu oraz wdrożeniem prototypu systemu cyberbezpieczeństwa o wadze państwowej.

Spis treści

1 Informacje ogólne o habilitancie

W swoich badaniach od lat koncentruję się na zagadnieniach związanych z teoretycznymi oraz praktycznymi aspektami projektowania algorytmów sterowania i analizy danych dla złożonych systemów teleinformatycznych. Wyniki mojej działalności naukowej w obszarze badań podstawowych (teoria sterowania, teoria gier oraz teoria przetwarzania sygnałów) znalazły bezpośrednie zastosowanie w teleinformatycznych systemach cyberbezpieczeństwa, zaprojektowanych przeze mnie oraz wdrożonych pod moim kierownictwem, które swoim swoim działaniem obejmują znaczącą część cyberprzestrzeni Polski.

W marcu 2006 roku rozpocząłem pracę w Pracowni Sterowania Siecią Pionu Naukowego NASK Państwowego Instytutu Badawczego. W swojej pracy badawczej koncentrowałem się wówczas na zagadnieniach związanych z projektowaniem punktów równowagi Nasha w grach niekooperacyjnych. Badania te umotywowane były potrzebą konstrukcji mechanizmów alokacji zasobów TCP/AQM, które byłyby odporne na działania (ataki) potencjalnie degradujące jakość usług w sieciach IP. Efekty ówczesnych eksperymentów zawarłem w rozprawie doktorskiej pt. *Coordination in hierarchical systems with rational agents*, którą napisałem pod opieką naukową prof. Krzysztofa Malinowskiego i obroniłem z wyróżnieniem w 2010 roku na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej. W roku 2010 objąłem stanowisko adiunkta w Pracowni Sterowania Siecią NASK.

W latach 2011-2013 byłem wykonawcą projektu ECONET (FP7), w ramach którego zaprojektowałem system energooszczędnego predykcyjnego sterowania konfiguracją urządzeń sieciowych. Koncepcja ta doczekała się fizycznego prototypu.

W 2014 roku objąłem stanowisko adiunkta w Zakładzie Sterowania i Systemów Instytutu Automatyki i Informatyki Stosowanej. W ramach działalności dydaktycznej miałem przyjemność opracować autorski program wykładu, ćwiczeń, zadań projektowych oraz zajęć laboratoryjnych, które obejmowały kluczowe z punktu widzenia praktycznych zastosowań elementy teorii układów dynamicznych, teorii stabilności oraz teorii projektowania układów sterowania. Za opracowanie nowego wykładu poświęconego teorii sterowania otrzymałem indywidualną III stopnia JM Rektora PW za osiągnięcia dydaktyczne w roku akademickim 2014/2015. Natomiast w roku 2017 otrzymałem nagrodę zespołową I stopnia stopnia JM Rektora PW za osiągnięcia naukowe (w latach 2015-2016), związane z rozwojem algorytmów i mechanizmów energooszczędnego sterowania w systemach teleinformatycznych.

Jestem także laureatem i wykonawcą grantu NCN w ramach konkursu OPUS-9, poświęconego rozwojowi mechanizmów energooszczędnego sterowania w systemach komputerowych dla obliczeń wielkiej skali.

W 2017 roku objąłem stanowisko kierownika Zakładu Inżynierii Systemów Informatycznych NASK Państwowego Instytutu Badawczego, gdzie mam możliwość koordynować projekt, rozwój oraz wdrożenie systemów informatycznych wczesnego wykrywania, monitorowania i reagowania na incydenty naruszające bezpieczeństwo teleinformatyczne. Systemy te swoim działaniem obejmują sieci infrastruktury krytycznej oraz administracji publicznej.

Obecnie w swoich badaniach naukowych koncentruję się na zaawansowanych zagadnie-

niach teoretycznych związanych z projektowaniem i implementacją mechanizmów adaptacyjnej filtracji oraz adaptacyjnego sterowania w systemach teleinformatycznych. Interesują mnie również zagadnienia uczenia maszynowego oraz zagadnienia związane z optymalizacją alokacji zasobów w klastrach obliczeniowych. Wyniki moich własnych badań naukowych oraz badań prowadzonych pod moim kierownictwem znalazły jak dotąd zastosowanie w jądrze systemu operacyjnego Linux, a także w systemach cyberbezpieczeństwa.

Jestem członkiem Rady Naukowej NASK Państwowego Instytutu Badawczego, pełnię także funkcję członka Komitetu Sterującego w ramach projektu ICT COST Action IC1406: High-Performance Modelling and Simulation for Big Data Applications.

1.1 Posiadane stopnie naukowe

Doktor nauk technicznych (2010)

Dziedzina: informatyka, specjalność: teoria gier
Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych
Tytuł rozprawy: Coordinatiton in hierarchical systems with rational agents
Promotor: prof. dr hab. inż. Krzysztof Malinowski
(rozprawa uznana przez Radę Wydziału EiTI PW za wyróżniającą się)

Magister inżynier (2005)

Zakres: Systemy informatczne wspomaagnia decyzji
Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych
Tytuł pracy: Operacyjne zarządzanie współpracą systemów autonomicznych w sieci Internet
Opiekun: prof. dr hab. inż. Krzysztof Malinowski
(studia ukończone z wynikiem celującym)

1.2 Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych

NASK Państwowy Instytut Badawczy

Adiunkt, kierownik Zakładu Inżynierii Systemów Informatycznych
Pion Badań i Rozwoju NASK PIB
ul. Kolska 12, 01-045 Warszawa
- od marca 2006: asystent
- od marca 2010: adiunkt

Politechnika Warszawska

Adiunkt, Zakład Sterowania Systemów, Zespół Złożonych Systemów
Instytut Automatyki i Informatyki Stosowanej
Wydział Elektroniki i Technik Informacyjnych
ul. Nowowiejska 15/19, 00-665 Warszawa
- od marca 2014

Marquette University
Visiting researcher
1250 W. Wisconsin Avenue
Milwaukee, WI 53233
- luty 2004

1.3 Charakterystyka działalności naukowo-badawczej

W swojej działalności naukowej koncentruję się na zagadnieniach **syntezy mechanizmów sterowania i przetwarzania danych dla złożonych systemów informatycznych**. Badania prowadzę w dwóch obszarach nauk podstawowych, a mianowicie w obszarze teorii gier i optymalizacji, oraz w obszarze teorii sterowania i przetwarzania sygnałów. Szczególnie istotne dla moich badań są zagadnienia związane z podejmowaniem decyzji w warunkach niepewności i asymetrii informacyjnej oraz zagadnienia sterowaniem procesami niestacjonarnymi. Efekty moich prac zostały zawarte w cyklu publikacji naukowych opisujących koncepcje i rozwiązania, które znalazły zastosowanie w dwóch systemach cyberbezpieczeństwa obejmujących swoim działaniem obszar całego kraju.

Projektowanie punktów równowagi Nasha

W pierwszym ze wspomnianych obszarów badawczych sformułowałem szereg twierdzeń opisujących proces projektowania punktów równowagi Nasha dla gier niekooperacyjnych indukowanych przez warunki optymalności Karusha-Kuhna-Tuckera. Znajomość warunków optymalności pozwala racjonalnym i autonomicznym decydentom na manipulację wartościami mnożników Lagrange'a, które koordynują proces współzawodnictwa.

Za najważniejsze uważam sformułowane przeze mnie warunki wystarczające implementacji w punkcie równowagi Nasha (wspomnianego typu gier) rozwiązań wypukłych zadań optymalizacji. Określiłem również relacje pomiędzy punktami równowagi Nasha w grach indukowanych przez dwa podstawowe typy mechanizmów alokacji zasobów z wyceną jednolitą, które obecnie są wykorzystywane m.in. w mechanizmach przeciwprzeciążeniowych protokołu TCP oraz przetargach publicznych realizowanych metodą aukcji otwartej (z licytacją). Efekty moich prac można również wykorzystać w projektowaniu algorytmów regulacji dla sieci TCP/AQM oraz analizie punktów równowagi tych niezwykle złożonych stochastycznych systemów dynamicznych. Przede wszystkim jednak wyniki te, wraz z wynikami dotyczącymi agregacji relacji porządkujących (uzyskanymi wcześniej w ramach rozprawy doktorskiej), znalazły **bezpośrednie zastosowanie w zaprojektowanych przeze mnie mechanizmach detekcji i mitygacji ataków DDoS**, które zostały wykorzystywane w systemie `FLDX`.

Charakteryzacja optymalnej polityki sterowania procesorem

Wyniki jakie uzyskałem w drugim ze wspomnianych obszarów opisują strukturę polityki sterowania procesorem oraz strukturę systemu koordynującego pracę procesorów w syste-

mie klastrze obliczeniowym. Badania przeprowadzone w tym obszarze związane były z problematyką energooszczędnego sterowania systemami teleinformatycznymi, takimi jak sieci TCP/IP oraz klastry maszyn wirtualnych. Obejmowały one zagadnienia syntezy polityki sterowania komponentami systemu, dynamicznej optymalizacji polityki routingu (pakietów lub zadań obliczeniowych) oraz prognozowania obciążenia komponentów systemu. Pierwsza grupa wyników moich badań opisuje strukturę polityki sterowania procesorem CPU w środowisku jądra systemu Linux. Polityka ta optymalizuje na horyzoncie nieskończonym dwukryterialny wskaźnik jakości uwzględniający koszty zużycia energii elektrycznej oraz wydajność obliczeniową procesora. Za pomocą sformułowanych przeze mnie twierdzeń charakteryzujących optymalną strukturę polityki sterowania możliwe jest przeprowadzenie oceny jakości projektu algorytmu sterowania energooszczędnego, oraz wskazanie obszaru w przestrzeni sterowań dopuszczalnych, w którym powinna znaleźć się polityka sterowania. Na podstawie tych wyników pod moim kierownictwem zaimplementowana została rodzina energooszczędnych sterowników CPU dla jądra systemu Linux. Badania eksperymentalne dowiodły, że w porównaniu ze standardowym energooszczędnym sterownikiem CPU **nowe sterowniki redukują zużycie energii elektrycznej oraz podnoszą wydajność usług systemowych**. Doświadczenia zgromadzone podczas pracy nad tym zagadnieniem okazały się mieć szersze zastosowanie - z powodzeniem wykorzystałem je również w projekcie mechanizmów kształtowania przepływów sieciowych zaburzanych przez atak typu DDoS.

Druga grupa wyników opisuje politykę predykcyjnego energooszczędnego sterowania konfiguracją routingu. Konfiguracja ta jest rozwiązaniem zadania optymalizacji przepływów w grafie (z routingiem wielościeżkowym opisanym macierzą incydencji w modelu łącze-ścieżka) przy ograniczeniach zadanych topologią sieci oraz prognozowanym natężeniu przepływów na łączach. Sformułowane zadanie pozwala szybko obliczać dostatecznie efektywną (chociaż suboptymalną) konfigurację aktywnych ścieżek w grafie oraz identyfikować urządzenia zdolne do obsługi bieżącego i prognozowanego przepływu zadań obliczeniowych. Tak sformułowane zadanie pozwala wyznaczyć optymalny poziom aktywności węzłów sieci lub alokację zasobów maszyn fizycznych obsługujących klaster maszyn wirtualnych. Może także znaleźć zastosowanie w systemach tłumienia ataków sieciowych, w których istnieje potrzeba wyznaczenia alternatywnych ścieżek routingu pakietów. Badania laboratoryjne dowiodły skuteczności zaprojektowanej i zaimplementowanej pod moim kierownictwem struktury sterowania.

Prognozowanie natężenia ruchu sieciowego

Do trzeciej grupy wyników zaliczam projekt oraz implementację algorytmów prognozowania natężenia ruchu sieciowego. Prace te zaowocowały projektem nowego algorytmu predykcji wykorzystującego ułamkowy i niestacjonarny model procesu dyfuzji, który umożliwia skuteczne prognozowanie szeregów czasowych o zmiennej charakterystyce. Algorytm ten oblicza krótkookresową prognozę wzrostów lub spadków natężenia ruchu na podstawie wyników testu hipotezy opisującej zależność pomiędzy dynamiką ułamkowego rzędu równania dyfuzji a wartością oczekiwaną prognozowanego procesu stochastycznego. Badania laboratoryjne pokazały,

że zaprojektowany algorytm prognozowania jest skuteczny. **Dlatego też algorytm ten został również wykorzystany w sieci akademickiej NASK PIB do prognozowania ataków DDoS i obciążenia łączy sieciowych**.

2 Opis zasadniczego osiągnięcia habilitacyjnego

2.1 Przedmiot osiągnięcia naukowego

Za podstawę do ubiegania się o stopień doktora habilitowanego w dziedzinie nauk technicznych w dyscyplinie informatyka przedstawiam osiągnięcie naukowe, na które składają się:

I CZĘŚĆ TEORETYCZNA

Jednotematyczny cykl publikacji pod tytułem: **Opracowanie podstaw teoretycznych syntezy wybranych mechanizmów sterowania dla systemów teleinformatycznych**;

II CZĘŚĆ PRAKTYCZNA

Praktyczne zastosowanie przedstawionych w cyklu publikacji wniosków w formie projektu architektury, implementacji i wdrożenia systemu cyberbezpieczeństwa FLDX. Jest to adaptacyjny system ochrony sieci przed wolumetrycznymi atakami typu DDoS, obejmujący swoim działaniem Ogólnopolską Sieć Edukacyjną (łączącą 19 000 szkół) oraz szkielet sieci NASK S.A. (o przepustowości 100 Gbps).

2.2 Wykaz prac będących podstawą wniosku habilitacyjnego

[N1] **Michał P. Karpowicz [70%]**, Piotr Arabas [20%], Ewa Niewiadomska-Szynkiewicz [10%]. Design and implementation of energy-aware application-specific CPU frequency governors for the heterogeneous distributed computing systems. *Future Generation Computer Systems*, 78:302–315, 2018.
[$\mathbf{IF}_{2018}$ **= 4.639**, 35 pkt. MNiSW]

[N2] **Michał P. Karpowicz [100%]**. Energy-efficient CPU frequency control for the Linux system. *Concurrency and Computation: Practice and Experience*, 28(2):420–437, 2016.
[$\mathbf{IF}_{2016}$ **= 1.114**, 20 pkt. MNiSW]

[N3] **Michał P. Karpowicz [70%]**, Piotr Arabas [20%], Ewa Niewiadomska-Szynkiewicz [10%]. Energy-aware multilevel control system for a network of Linux software routers: design and implementation. *Systems Journal, IEEE*, PP(99):1–12, 2015.
[$\mathbf{IF}_{2015}$ **= 2.114**, 35 pkt. MNiSW]

[N4] **Michał P. Karpowicz [100%]**. Nash equilibrium design and price-based coordination in hierarchical systems. *International Journal of Applied Mathematics and Computer Science*, 22(4):951–969, 2012.
[$\mathbf{IF}_{2012}$ **= 1.008**], 25 pkt. MNiSW]

[N5] **Michał P. Karpowicz [80%]**, Krzysztof Malinowski [20%]. Price-based coordinability in hierarchical systems with information asymmetry. *Control and Cybernetics*, 42(3):85–110, 2013.
[14 pkt. MNiSW]

[N6] **Michał P. Karpowicz [100%]**. On the design of the TCP/AQM traffic flow control mechanisms. *Journal of Telecommunications and Information Technology*, 4:31–37, 2012.
[14 pkt. MNiSW]

[N7] Piotr Arabas [50%], **Michał P. Karpowicz [50%]**. Server power consumption: measurements and modeling with MSRs. In *Challenges in Automation, Robotics and Measurement Techniques*, pages 233–244. Springer, 2016.
WoS

[N8] Michał Getka [80%], **Michał P. Karpowicz [20%]**. Aspekty energooszczędnego sterowania wydajnością pracy procesora systemu komputerowego zgodnego z ACPI w systemie Linux. *Przegląd Elektrotechniczny*, 94, 2018.
[14 pkt. MNiSW]

[N9] **Michał P. Karpowicz [100%]**. Designing auctions: a historical perspective. *Journal of Telecommunications and Information Technology*, 3:114–122, 2011.
[12 pkt. MNiSW]

Powyższe publikacje powstały w latach 2010-2018, czyli w okresie po uzyskaniu przeze mnie z wyróżnieniem stopnia naukowego doktora nauk technicznych w dyscyplinie informatyka. Przy nazwiskach współautorów podałem ich szacunkowy udział procentowy w przygotowaniu publikacji. Udział jakościowy wnioskodawcy w powstanie prac o charakterze współautorskim został opisany w załączniku Z3, podpisanym przez współautorów.

Sumaryczny Impact Factor publikacji stanowiących osiągnięcie naukowe wynosi **8.875**, natomiast ważony **6.85**.

2.3 Motywacja

Zjawiska wynikające ze złożoności systemów od dawna wzbudzały moje zainteresowanie, a ich uniwersalny charakter sprawiał, że problemy pojawiające się w pracach nad sieciami teleinformatycznymi, klastrami maszyn wirtualnych czy systemami ekonomicznymi mogłem rozwiązać za pomocą rozszerzonej palety narzędzi, zbudowanej przeze mnie na potrzeby badania złożoności na gruncie teorii gier, sterowania i sygnałów.

Wśród wspomnianych zjawisk szczególnie ciekawa wydała mi się kwestia współzawodnictwa autonomicznych decydentów, którzy w warunkach niepewności i asymetrii informacyjnej dążą do realizacji indywidualnych i najczęściej wykluczających się celów, m.in. dotyczących wykorzystania współdzielonych zasobów systemowych. Zaprojektowanie mechanizmów, które pozwolą osiągnąć w takich warunkach zgodność celów pracy całego systemu z celami jego

komponentów, jest zadaniem niezwykle trudnym. **Obrana przeze mnie ścieżka badawcza pozwoliła mi jednak odkryć szereg konstrukcji matematycznych, które okazały się nie tylko satysfakcjonujące pod względem formalnym, lecz także znalazły zastosowanie w praktyce. Dowodem tego są udane wdrożenia systemów złożonych zapewniające cyberbezpieczeństwo na poziomie całego kraju: systemów `FLDX` oraz `ARAKIS`.**

Główne wyniki moich badań zostały bezpośrednio wykorzystane w systemie `FLDX`, czyli zaprojektowanym przeze mnie i wdrażanym pod moim kierownictwem przez NASK PIB adaptacyjnym systemie wczesnego wykrywania oraz automatycznego tłumienia wolumetrycznych ataków sieciowych typu DDoS. Obejmie on swoim działaniem Ogólnopolską Sieć Edukacyjną, publiczną sieć telekomunikacyjną zapewniającą szkołom dostęp do bezpiecznego internetu, oraz szkielet sieci NASK S.A. System `FLDX` jest wielowarstwowym uczącym się (adaptacyjnym) systemem sterowania wykrywającym i tłumiący zaburzenia ruchu sieciowego, zdolnym do obsługi ataków o natężeniu przekraczających 100 Gbps. Jego mechanizmy diagnostyczne automatycznie identyfikują oraz izolują źródła zaburzeń w czasie nieosiągalnym dotychczas na rynku systemów cyberbezpieczeństwa. Zaprojektowane przeze mnie algorytmy uczenia maszynowego (w trybie bez nadzoru) samodzielnie generują wiedzę na temat obserwowanych zjawisk sieciowych oraz zapisują ją w postaci reguł decyzyjnych wyrażonych w języku filtracji pakietów przełączników sieciowych. W ten sposób dostarcza on operatorom ośrodków NSOC (Network Security Operation Center) narzędzia pozwalające na szybkie i skuteczne odpieranie wielowymiarowych i zmiennych w czasie ataków.

Innowacyjność rozwiązania

Innowacyjność wykorzystanych w systemie `FLDX` rozwiązań wynika ze sprowadzenia zadań detekcji oraz tłumienia ataków DDoS do postaci dobrze matematycznie sformułowanych problemów z obszaru analizy sygnałowej, związanych z syntezą reguł filtracji, separacji i rekonstrukcji sygnałów, oraz odpowiednio określonych zagadnień z obszaru teorii sterowania, polegających na adaptacyjnej syntezie algorytmów prognozowania i tłumienia zaburzeń ruchu. Podejście to pozwala w rozważanym kontekście m.in. na:

- optymalną automatyzację procesów przetwarzania danych,
- zapewnienie odporności systemu na zmiany charakterystyki obserwowanych i kształtowanych procesów dynamicznych,
- minimalizacją czasu przetwarzania danych, wykonywania obliczeń i zmian konfiguracji urządzeń sieciowych (przełączników lub routerów),
- uzyskiwanie wyników działania systemu o gwarantowanych wskaźnikach jakości związanych ze skutecznością detekcji ataków oraz skutecznością kształtowania dynamiki przepływów sieciowych.

Architektura systemu wynika z wypracowanej przeze mnie w ciągu ostatnich lat metodyki prowadzenia prac badawczo-rozwojowych. Polega ona w dużym uproszczeniu na przekształceniu problemu inżynierskiego do postaci adekwatnego problemu matematycznego, a następnie wykorzystaniu właściwości jego rozwiązania w projekcie i implementacji algorytmu rozwiązującego postawiony problem inżynierski. Należy podkreślić, że konstruowany w tym procesie problem matematyczny powinien posiadać rozwiązanie w postaci efektywnego algorytmu, który nie wymaga prowadzenia wymagających numerycznie obliczeń. Szczególną rolę w moich pracach odgrywały zatem równania Karusha-Kuhna-Tuckera, równanie Bellmana oraz diofantyczne równanie syntezy algorytmów filtracji.

Podejście wykorzystujące zaawansowaną matematykę teorii sygnałów i sterowania w projektowaniu architektury informatycznego systemu cyberbezpieczeństwa nie było dotychczas stosowane w rozwiązaniach dostępnych na rynku, co uzasadniło objęcie konstrukcji systemu `FLDX` ochroną patentową. Badanie podmiotowe i przedmiotowe wykonane na zlecenie NASK PIB przez rzecznika patentowego wykazało, że istotne elementy rozwiązania nie pojawiały się jak dotąd w międzynarodowej bazie wniosków patentowych, zarówno tych rozpatrzonych jak i oczekujących na rozpatrzenie. W projekcie architektury systemu `FLDX` wykorzystane zostały wyniki badań przeprowadzonych przeze mnie w okresie ostatnich lat oraz zgromadzone w tym okresie doświadczenia praktyczne.

3 Struktura i opis teoretycznej części zasadniczego osiągnięcia naukowego

Wyniki moich prac tworzą podstawy teoretyczne syntezy mechanizmów sterowania przeznaczonych dla systemów sieciowych i teleinformatycznych. Składają się na nie zagadnienia z trzech obszarów.

1. Mechanizmy alokacji zasobów sieciowych

    - twierdzenia określające warunki i procedurę implementacji w punkcie równowagi Nasha rozwiazań wypukłych zadań optymalizacji (alokacji zasobów sieciowych),
    - twierdzenia charakteryzujące własności punktów równowai Nasha w grach definiowanych przez reguły alokacji zasobów z wyceną jednolitą,

2. Energooszczędne sterowanie systemem teleinformatycznym

    - twierdzenia charakteryzujące optymalną politykę enerooszczędnego sterowania procesorem,
    - opracowanie na bazie sformułowanych twierdzeń i modeli matematycznych architektury wielowarstwowego, adaptecyjnego systemu energooszczędnego sterowania klastrem obliczeniowym,
    - opracowanie, implementacja i wdrożenie formalnego modelu procesu detekcji i tłumienia ataku DDoS w postaci automatu skończonego
    - opracowanie, implementacja i wdrożenie hierarchicznej rodziny algorytmów detekcji ataków DDoS (w warstwach L2-L4) wykorzystującej autorską operację rekurencyjnej rekonstrukcji składowych ataku w odpowiednio zidentyfikowanej bazie sygnałów,

3. Synteza mechanizmów sterowania adaptacyjnego

    - opracowanie systemu sterowania adaptacyjnego tłumiącego ataki DDoS lub zaburzenia ruchu sieciowego o nieznanej dynamice, kształtującego dynamikę przepływów pakietów, identyfikującego modele dynamiki ataków, prognozującego natężenie ruchu sieciowego,
    - opracowanie matematycznego modelu reprezentacji wiedzy o ruchu sieciowym wyrażonego w języku reguł filtracji pakietów,
    - opracowanie reguł klasyfikacji przepływów sieciowych (algorytmy uczenia bez nadzoru).

3.1 Mechanizmy alokacji zasobów sieciowych

Rozważmy sieć TCP/IP (TCP/AQM) złożoną z E łączy wykorzystywanych przez strumienie pakietów wysyłanych przez S źródeł. Każde źródło $s = 1,...,S$ realizuje połączenie pomiędzy parą węzłów w rozważanej sieci. Połączenie takie może zostać zrealizowane na $P_s,\ s = 1,...,S$, ścieżkach zbudowanych z łączy sieci. Wielościeżkowy routing, dopuszczalny w rozważanym modelu, jest określony przez macierze (incydencji) $\mathbf{A}_s = [a_{spe}],\ s = 1,...,S$, takie, że $a_{spe} = 1$ jeżeli łącze e wykorzystywane jest przez ścieżkę p wychodzącą ze źródła s, oraz $a_{spe} = 0$ w przeciwnym wypadku.

Z każdym źródłem ruchu można związać wskaźnik jakości $U_s,\ s = 1,...,S$. Wartość wskaźnika $U_s(x_s)$ reprezentuje użyteczność osiąganej szybkości transmisji pakietów x_s dla źródła ruchu. Podobnie, z każdym łączem $e = 1,...,E$ można powiązać wskaźnik jakości C_e opisujący koszt $C_e(y_e)$ związany z obsługą natężenia $y_e > 0$ przepływu pakietów.

Procedura syntezy (projektowania) algorytmów sterujących szybkością transmisji i kolejkowania pakietów dla rozważanej sieci składa się z dwóch etapów. Pierwszy polega na wyborze punktu równowagi będącego rozwiązaniem wielokryterialnego zadania optymalizacji użyteczności sieci. Drugi etap polega na syntezie algorytmów sterowania, czyli dyskretnych układów dynamicznych, dla źródeł ruchu oraz łączy sieciowych (kolejek urządzeń sieciowych), które gwarantują zbieżność do wskazanego punktu równowagi. Sieć rozpatrywana jest tutaj zatem jako rozproszony stochastyczny układ dynamiczny złożony ze współzawodniczących ze sobą układów sterowania.

Podstawowe sformułowanie problemu optymalizacji użyteczności sieci, stanowiące punkt wyjścia dla procedury syntezy algorytmów sterowania oraz model preferencji projektanta systemu, ma następującą znaną i niezwykle uniwersalną postać [8, 9]:

$\texttt{SYSTEM}(\mathbf{U},\mathbf{c},\mathbf{A})$:

$$\left|\begin{array}{ll} \text{zmaksymalizuj} & \sum_s U_s(x_s) \\ \text{przy ograniczeniach} & \mathbf{A}^T\mathbf{x} \le \mathbf{c} \\ \text{względem} & x_s \ge 0,\ s = 1,...,S, \end{array}\right.$$

przy czym wektor $\mathbf{c}^T = [c_1,...,c_e]$ określa przepustowość łączy. Problem SYSTEM definiuje punkt odniesienia dla projektu algorytmów sterowania. Jak łatwo zauważyć, jest to problem optymalizacji wielokryterialnej z utylitarną agregacją modeli preferencji w postaci sumy funkcji użyteczności źródeł i łączy. Należy zaznaczyć, że w powyższym modelu $C_e(y_e) = 0$ dla $0 \le y_e \le c_e$ and $C_e(y_e) = +\infty$ dla $y_e > c_e$. Alternatywne sformułowanie zadania maksymalizacji użyteczności ma następującą postać:

$\texttt{SYSTEM}(\mathbf{U},\mathbf{C},\mathbf{A})$:

$$\left|\begin{array}{ll} \text{zmaksymalizuj} & \sum_s U_s(\sum_p x_{sp}) - \sum_e C_e(\sum_s \sum_p x_{sp} a_{spe}) \\ \text{względem} & x_{sp} \ge 0,\ s = 1,...,S,\ p = 1,...,P_s. \end{array}\right.$$

Dysponując specyfikacją punktu równowagi można przejść do etapu syntezy algorytmów sterowania. Załóżmy na potrzeby wstępnej ilustracji koncepcji metody syntezy, że źródło ruchu informuje sieć (określonym protokołem) sygnałem $\theta_s \geq 0$ o swojej gotowości do zapłaty za szybkość $x_s \geq 0$ przesyłania pakietów. Załóżmy również, że szybkość ta jest wyznaczana zgodnie z regułą alokacji:

$$x_s = \bar{\xi}_s(\theta_s, \lambda_s) = \begin{cases} \theta_s/\lambda_s, & \text{dla } \theta_s > 0; \\ 0, & \text{dla } \theta_s = 0, \end{cases} \tag{1}$$

gdzie $\lambda_s = \sum_e a_{se}\mu_e \geq 0$ jest ceną wykorzystania sieci równą sumie cen wykorzystania łączy budujących ścieżkę. Jednocześnie z tak obliczoną alokacją zasobów związana jest następująca reguła wyceny $w_s \geq 0$ alokacji:

$$w_s = \bar{\eta}_s(\theta_s, \lambda_s) = \theta_s. \tag{2}$$

W tym punkcie należy zauważyć, że ceny $\mu_e \geq 0,\ e = 1,...,E$, są mnożnikami Lagrange'a związanymi z ograniczeniami przepustowości łączy, co pozwala interpretować strukturę sterowania wyłaniającą się ze złożoności rozważanego systemu jako strukturę sterowania z koordynacją iteracyjna metodą cen [16]. Mnożniki Lagrange'a pełnią w tym układzie sterowania rolę sygnału sprzężenia zwrotnego, który informuje źródła ruchu o poziomie obciążenia sieci przez sumaryczny przepływ pakietów.

Poszukiwany jest oczywiście algorytm wyznaczania sygnału $\theta_s \geq 0$, ujawniający sieci preferencje źródła ruchu $s = 1,...,S$. Algorytm ten można skonstruować na podstawie rozwiązania następującego zadania optymalizacji [8, 9]:

$\texttt{USER}_s(U_s, \lambda_s)$:

zmaksymalizuj	$U_s(\theta_s/\lambda_s) - \theta_s$
względem	$\theta_s \geq 0.$

Konstrukcja algorytmu opiera się na następującym rozumowaniu. Złóżmy, że cena λ_s jest ustalona a źródło ruchu nie może na tę cenę wpływać. Przyjmując to założenie można wykazać, że sieć dysponująca informacją w postaci wektora $\boldsymbol{\theta} = (\theta_1, ..., \theta_S)$, może wyznaczyć przepływy $x_s \geq 0,\ s = 1,...,S$, przy pomocy algorytmów kolejkowania pakietów (AQM) oraz sterowania szerokością okna transmisji pakietów (TCP) rozwiązujących w sposób asynchroniczny i zdecentralizowany zadanie:

$\texttt{NETWORK}(\mathbf{A}, \mathbf{c}, \boldsymbol{\theta})$:

zmaksymalizuj	$\sum_s \theta_s \log(x_s)$
przy ograniczeniach	$\mathbf{A}^T\mathbf{x} \leq \mathbf{c}$
względem	$x_s \geq 0,\ s = 1,...,S.$

Algorytm sterowania szybkością transmisji pakietów jest wobec tego wariantem algorytmu optymalizacji, sparametryzowanego sygnałem θ_s przez każde źródło ruchu. W postaci ciągłej ma on postać następującego układu równań różniczkowych:

$$\begin{aligned}&\texttt{RATE}(\mathbf{A},\mathbf{q},\boldsymbol{\theta}):\\ &\left|\begin{array}{l}\dfrac{\mathrm{d}x_s(t)}{\mathrm{d}t}=\kappa\left[\theta_s-x_s(t)\sum_e a_{se}\mu_e(t)\right],\ s=1,\ldots,S,\\ \mu_e(t)=q_e(\sum_s a_{se}x_s(t)),\ e=1,\ldots,E.\end{array}\right.\end{aligned}$$

Jak wynika z przedstawionego rozumowania ceny $\mu_e \geq 0,\ e=1,\ldots,E$, wyznaczane są tutaj przez algorytmy kolejkowania przekazujący do sieci informację o krańcowym koszcie obciążenia łącza. Sygnał θ_s powinien być natomiast wyznaczany na podstawie warunków optymalności pierwszego rzędu dla rozwiązania zadania $\texttt{USER}_s(U_s,\lambda_s)$:

$$\theta_s(t)=x_s(t)U_s'(x_s(t)),\ s=1,\ldots,S, \tag{3}$$

gdzie $U_s' \equiv \mathrm{d}U_s/\mathrm{d}x_s$. Szczegóły dotyczące implementacji powyższego algorytmu można znaleźć m.in. w pracach [14, 11, 17, 16, 13, 15, 18].

Powyższe rozumowanie bazuje na założeniu, że cena λ_s jest ustalona a źródło ruchu nie może na tę cenę wpływać. W warunkach asymetrii informacyjnej, w naturalny sposób występujących w systemach złożonych, takie założenie może nie być spełnione. Użytkownicy sieci (administratorzy źródeł ruchu) znają zasady alokacji zasobów sieciowych, jest to wiedza powszechna, jednak posiadają oni prywatną wiedzę na temat swojego modelu preferencji (funkcji użyteczności). Ponieważ informacja na temat użyteczności, wyrażona sygnałem θ_s, stanowi podstawę dla alokacji zasobów, a w szczególności wyznaczenia cen $\mu_e \geq 0,\ e=1,\ldots,E$, użytkownicy sieci mogą podejmować próby manipulowania procesem alokacji. W rozważanym modelu sprowadza się to do wpływania na sygnał sprzężenia zwrotnego λ_s poprzez zastosowanie strategii spekulatywnych. Strategie te polegają na ujawnianiu do sieci informacji o celowo zmodyfikowanym modelu preferencji, różniącym się od rzeczywistego.

W swoich badaniach **scharakteryzowałem konstrukcje strategii spekulatywnych prowadzących do wyników korzystnych z punktu widzenia pojedynczego źródła ruchu (a raczej jego administratora) w grach indukowanych przez mechanizmy alokacji i jednolitej wyceny zasobów**. Udowodniłem, że optymalna strategia spekulatywna dla gry zdefiniowanej regułami $(\bar{\xi},\bar{\eta})$ może polegać na zaniżaniu wartości sygnału θ_s. Zgodnie z nią źródło s jest motywowane do komunikowania sygnału:

$$\theta_s(t)=x_s(t)\tilde{U}_s'(x_s(t)), \tag{4}$$

dla odpowiednio skonstruowanej funkcji $\tilde{U}_s$, gdzie $\tilde{U}_s'(x_s(t)) < U_s'(x_s(t))$.

Konsekwencje wykorzystania strategii spekulatywnych w grach indukowanych mechanizmami (aukcjami) z wyceną jednolitą opisałem w pracy [N5]. Niech $\boldsymbol{\Theta}=\Theta_1\times\cdots\times\Theta_n$ oznacza

przestrzeń sygnałów wytwarzanych przez źródło ruchu. Mechanizm jest iloczynem kartezjańskim funkcji $m_i : \boldsymbol{\Theta} \to X \times W,\ i = 1,...,n$, takich, że:

$$m_i(\boldsymbol{\theta}) = (\xi_i(\boldsymbol{\theta}), \eta_i(\boldsymbol{\theta})),\ i = 1,...,n. \tag{5}$$

Funkcja $\xi_i : \boldsymbol{\Theta} \to X$ oznacza regułę alokacji zasobu, natomiast $\eta_i : \boldsymbol{\Theta} \to W$ regułę wyceny.

Reguły mechanizmu są narzędziami koordynacji działań autonomicznych decydentów zdolnych do prowadzenia działań spekulatywnych. Indukują one grę, której rozwiązania nie są w ogólnym przypadku tożsame z rozwiązaniami zadania `SYSTEM`. Moje badania skoncentrowane były zatem na warunkach, jakie muszą być spełnione przez reguły mechanizmu, aby rozwiązanie wspomnianego problemu mogło być osiągalne w punkcie równowagi Nasha. W punkcie równowagi Nasha $\boldsymbol{\theta}^*$ gry z maksymalizowanymi wypłatami $J_i,\ i = 1,...,n$, jest określony przez układ nierówności:

$$J_i((\theta_i^*, \boldsymbol{\theta}_{-i}^*)) \geq J_i((\theta_i, \boldsymbol{\theta}_{-i}^*)),\ \theta_i \in \Theta_i,\ i = 1,...,n, \tag{6}$$

gdzie przyjęta jest notacja $\boldsymbol{\theta}_{-i} = (\theta_1,...,\theta_{i-1},\theta_{i+1},...,\theta_n^*)$ oraz $\boldsymbol{\theta} \equiv (\theta_i^*, \boldsymbol{\theta}_{-i}^*),\ i = 1,...,n$. Swoje badania przeprowadziłem przy założeniu, że każdy decydent przesyła do mechanizmu sygnał będący rozwiązaniem zadania:

`AGENT`$_i(m_i)$:

$$\left|\begin{array}{ll} \text{zmaksymalizuj} & J_i(m_i(\boldsymbol{\theta})) = U_i(\xi_i(\boldsymbol{\theta})) - \eta_i(\boldsymbol{\theta}) \\ \text{względem} & \theta_i \geq 0. \end{array}\right.$$

Przyjmując powyższe założenie **odkryłem szereg niezwykle ciekawych własności punktów równowagi Nasha** gier niekooperacyjnych zdefiniowanych przez mechanizm typu *payment-bidding*:

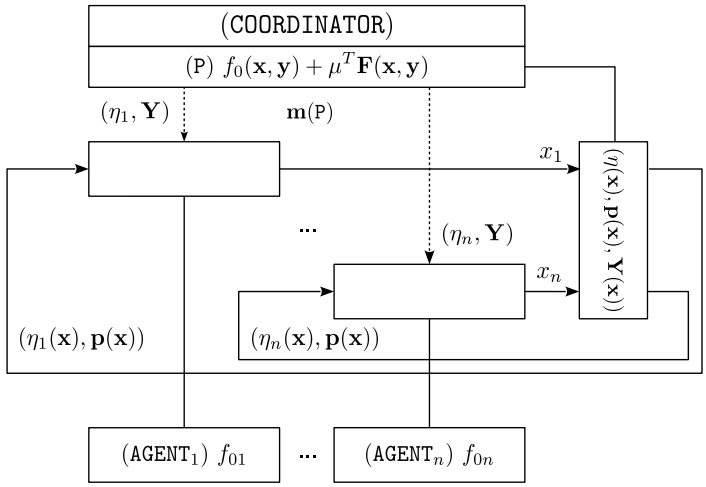
`PBM`$(\boldsymbol{\theta})$:

$$\left|\begin{array}{l} \hat{\xi}_i(\boldsymbol{\theta}) = \begin{cases} \theta_i / p(\hat{y}(\boldsymbol{\theta})), & \text{if } \theta_i > 0; \\ 0, & \text{if } \theta_i = 0, \end{cases} \\ \hat{\eta}_i(\boldsymbol{\theta}) = \theta_i \\ \sum_{i=1}^{n} \hat{\eta}_i(\boldsymbol{\theta}) = \hat{y}(\boldsymbol{\theta}) p(\hat{y}(\boldsymbol{\theta}), \end{array}\right.$$

oraz mechanizm typu *demand-bidding*:

`DBM`$(\boldsymbol{\theta})$:

$$\left|\begin{array}{l} \tilde{\xi}_i(\boldsymbol{\theta}) = \theta_i, \\ \tilde{\eta}_i(\boldsymbol{\theta}) = \theta_i p(\tilde{y}(\boldsymbol{\theta})) \\ \tilde{y}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \tilde{\xi}_i(\boldsymbol{\theta}). \end{array}\right.$$



Rysunek 1: Architektura mechanizmu koordynacji i alokacji zasobów [N4, N6].

Mechanizmy stanowią ogólny model reguł alokacji i jednolitej wyceny zasobów podzielnych wykorzystywanych w wielu systemach złożonych, w tym sieciach TCP/IP, klastrach maszyn wirtualnych oraz systemach ekonomicznych. Sformułowane przeze mnie w pracy [N5] twierdzenia pokazują, że w przypadku gier z wyceną jednolitą indukowanych przez powyższe mechanizmy:

- każdy gracz komunikuje zredukowany popyt na zasoby sieciowe, co prowadzi do obniżonej ceny równowagi i redukcji poziomu wykorzystania łączy;
- część graczy może otrzymać wyższe poziomy alokacji oraz poprawić swoją wypłatę z gry;
- część graczy poprawić wypłatę z gry przy obniżonej alokacji zasobów.

W warunkach asymetrii informacyjnej mechanizmy z wyceną jednolitą nie mogą zatem w ogólnym przypadku prowadzić do osiągnięcia w punktach równowagi Nasha rozwiązań zadania SYSTEM. W systemach złożonych, w których występują autonomiczni decydenci posiadający prywatną wiedzę, należy oczekiwać rozbieżności celów indywidualnych z celami systemowymi. Może to prowadzić do rozwiązań, które nie są Pareto-optymalne z punktu widzenia systemu jako całości. Wniosek ten **poszerza zbiór wyników opisujących właściwości punktów równowagi Nasha**, m.in. zawartych w pracach [19, 6].

Kolejne pytanie, na które poszukiwałem odpowiedzi dotyczyło **możliwości skonstruowania reguł gry generującej w punkcie równowagi Nasha rozwiązania wielokryterialnego problemu optymalizacji zasów sieciowych**, tzn. zadania SYSTEM. W pracy [N4, N6] udowodniłem przytoczone poniżej twierdzenie.

Twierdzenie 1. *Rozważmy problem:*

$$\begin{aligned}&\mathtt{P}(f_j,\ j=0,...,m):\\&\left|\begin{array}{ll}\textit{zminimalizuj} & f_0(\mathbf{x},\mathbf{y})\ \textit{względem}\ (\mathbf{x},\mathbf{y})\in\mathbb{R}^n\times\mathbb{R}^m\\ \textit{przy ograniczeniach} & f_j(\mathbf{x},\mathbf{y})=0\ \textit{dla}\ n,m>0,\ f_j\in\mathscr{C}^2,\ j=0,...,m.\end{array}\right.\end{aligned}$$

Przyjmijmy, że $f_j(\mathbf{x},\mathbf{y})=\sum_i f_{ji}(x_i)+g_j(\mathbf{y})$ *dla* $j=0,...,m$. *Niech* $\bar{\mathbf{z}}=(\bar{\mathbf{x}},\bar{\mathbf{y}})$ *będzie punktem, w którym spełnione są warunki konieczne optymalności pierwszego i drugiego rzędu oraz* $\det\nabla_{\mathbf{y}}\mathbf{F}(\bar{\mathbf{x}},\bar{\mathbf{y}})\neq 0$. *Załóżmy, że dla* $\mathbf{x}\in\mathbb{B}_\epsilon(\bar{\mathbf{x}})=\{\mathbf{x}\colon\|\bar{\mathbf{x}}-\mathbf{x}\|\leq\epsilon,\ \epsilon>0\}$ *funkcje* $\eta_i,\ i=1,...,n$, *są określone przez układ równań:*

$$\frac{\partial\eta_i(\mathbf{x})}{\partial x_i}=\sum_j p_j(\mathbf{x})\frac{\partial f_{ji}(x_i)}{\partial x_i},\ i=1,...,n, \tag{7}$$

gdzie

$$\mathbf{p}(\mathbf{x})\equiv-\left(\frac{\partial\mathbf{F}^T}{\partial\mathbf{y}}(\mathbf{x},\mathbf{Y}(\mathbf{x}))\right)^{-1}\frac{\partial f_0}{\partial\mathbf{y}}(\mathbf{x},\mathbf{Y}(\mathbf{x}))\ \textit{oraz}\ \mathbf{F}(\mathbf{x},\mathbf{Y}(\mathbf{x}))\equiv 0. \tag{8}$$

Załóżmy również, że dla $\mathbf{v}\in\mathbb{R}^n\setminus\{0\}$ *mamy:*

$$\mathbf{v}^T\frac{\partial\mathbf{Y}^T}{\partial\mathbf{x}}(\bar{\mathbf{x}})\frac{\partial^2 H}{\partial\mathbf{y}^2}(\bar{\mathbf{x}},\mathbf{p}(\bar{\mathbf{x}}))\frac{\partial\mathbf{Y}}{\partial\mathbf{x}^T}(\bar{\mathbf{x}})\mathbf{v}>0, \tag{9}$$

przy czym $H(\mathbf{x},\mathbf{p}(\mathbf{x}))=f_0(\mathbf{x},\mathbf{Y}(\mathbf{x}))+\mathbf{p}(\mathbf{x})^T\mathbf{F}(\mathbf{x},\mathbf{Y}(\mathbf{x}))$. *Wówczas* $\bar{\mathbf{z}}$ *jest rozwiązaniem problemu* P *oraz rozwiązaniem problemów:*

$$\begin{aligned}&\mathtt{PAYOFF}_i(f_{0i},\bar{x}_i,\mathbf{x}_{-i}),\ i=1,...,n:\\&\left|\begin{array}{ll}\textit{zminimalizuj} & J_i(x_i,\mathbf{x}_{-i})=f_{0i}(x_i)+\eta_i(\mathbf{x})\\ \textit{względem} & x_i\in\mathbb{B}_\epsilon(\bar{x}_i),\end{array}\right.\end{aligned}$$

dla $\mathbf{x}_{-i}=(x_1,...,x_{i-1},x_{i+1},...,x_n)$.

Twierdzenie to pokazuje, że **rozwiązanie problemu P można zaimplementować w punkcie Nasha gry o odpowiednio zaimplementowanych wypłatach** $J_i,\ i=1,...,n$, **jednak odbywa się to kosztem części graczy lub projektanta mechanizmu**.

Procedura projektowania reguł mechanizmu $\mathbf{m}(\mathtt{P})$, będąca wnioskiem z twierdzenia i zilustrowana na Rysunku 1, zakłada wykorzystanie lokalnych własności rozwiązania równania $\mathbf{F}(\mathbf{x},\mathbf{Y}(\mathbf{x}))\equiv 0$ w otoczeniu rozwiązania problemu P. Procedura określa rodzinę gier koordynujących interakcje graczy przy pomocy reguły $\mathbf{y}=\mathbf{Y}(\mathbf{x})$ oraz reguł alokacji i wyceny niejednolitej (dyskryminującej) $(\xi_i,\eta_i),\ i=1,...,n$. Konstrukcja reguł sprawia, że warunki optymalności pierwszego rzędu zadań lokalnych $\mathtt{PAYOFF}_i,\ i=1,...n$, są równoważne warunkom optymalności pierwszego rzędu zadania P.

Przeprowadzone przeze mnie analizy pozwalają także **porównać koszty wprowadzenia do systemu mechanizmów opisanych twierdzeniem z kosztami systemowymi wynikającymi ze stosowania przez graczy strategii spekulatywnych w grach z wyceną jednolitą**. Zaskakujące wnioski płynące z tej analizy są następujące [N5, N4, N9]:

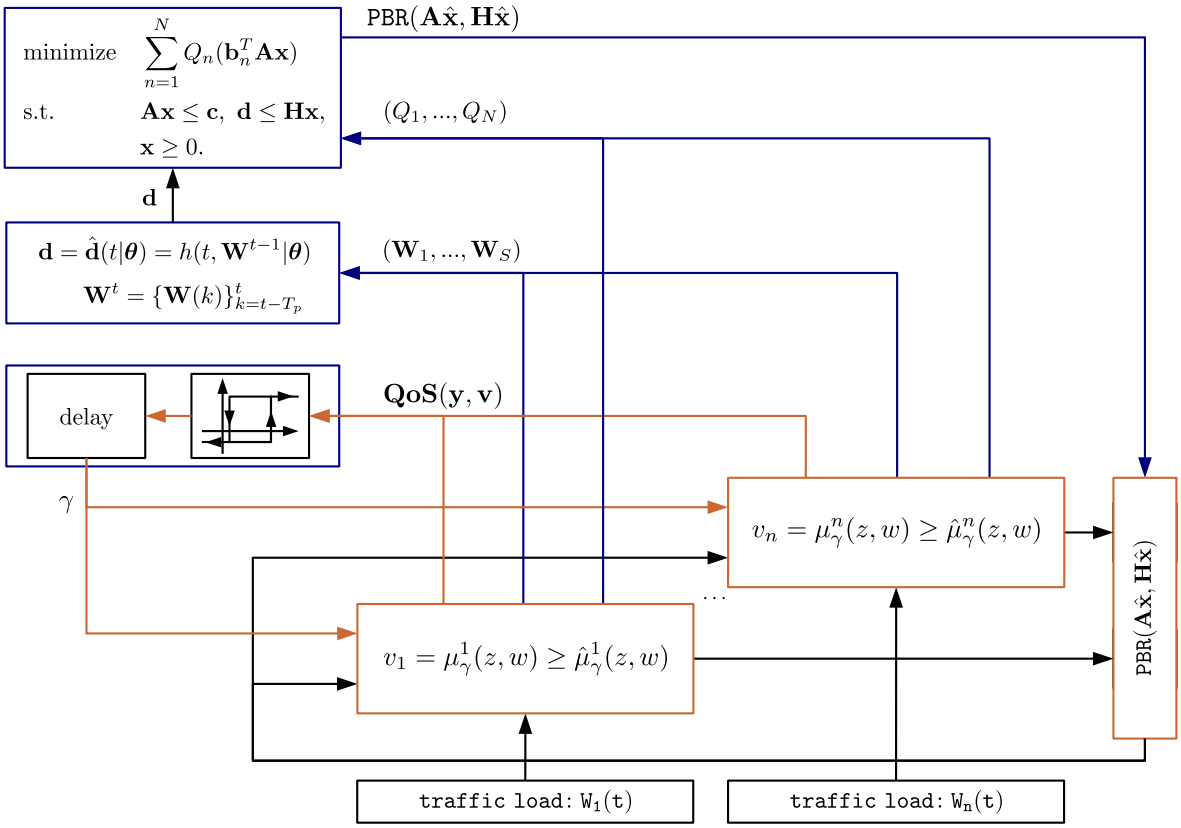
- w warunkach asymetrii informacyjnej strategie spekulacyjne mogą generować koszty systemowe, które są niższe od tych wynikających z wprowadzenia opłat niejednolitych,
- wybór zmiennych zależnych $\mathbf{y} = \mathbf{Y(x)}$ (np. współdzielonych łączy w sieci), kształtowanych przez zmienne niezależne $\mathbf{x}$, ma znaczenia dla efektywności równowagi Nasha,
- najsłabsi gracze mogą stracić podejmując konkurencję z silniejszymi graczami.

Omówione powyżej wyniki mają duże znaczenie praktyczne dla projektowania mechanizmów alokacji zasobów i reguł koordynacji współzawodnictwa autonomicznych decydentów w systemach złożonych. **Poszerzają one katalog twierdzeń opisujących właściwości punktów równowagi Nasha oraz dostarczają wskazówek dotyczących zasad ich projektowania**. Opisują one konstrukcje gier będących rozwiązaniem problemu sprawiedliwego (w sensie określonym metodą agregacji preferencji) podziału zasobów systemowych. Właśnie w tym kontekście, zagadnienia podziału zasobów sieciowych, powyższe wyniki znalazły bezpośrednie zastosowanie w zaprojektowanym przeze mnie systemie `FLDX` ochrony łączy warstw L2-4 przed atakami typu DDoS. Mechanizmy te są **wykorzystywane w procesie zarządzania stanem przepływu monitorowanego** jako test naruszania przez przepływ ograniczeń dotyczących bezpiecznego współdzielenia łącza chronionego.

3.2 Energooszczędne sterowanie systemem teleinformatycznym

Głównym osiągnięciem moich badań nad energooszczędnym sterowaniem systemem sieciowym lub obliczeniowym, opisanym w pracach [N1, N2, N7, N3], jest **opracowanie teoretycznej koncepcji wielowarstwowego systemu sterowania umożliwiającego redukcję zużycia energii** w sieci serwerów pracujących na systemie operacyjnym Linux i obsługujących przepływy pakietów. Zaproponowana struktura zapewnia sterowanie w czasie rzeczywistym, które dynamicznie modyfikuje konfigurację sieci ze względu na krótkoterminowe prognozy obciążenia sieci. Zadanie projektowe polegało tu przede wszystkim na odpowiednim sformułowaniu problemu optymalizacji sterowania. Po pierwsze zaproponowałem relaksację problemu optymalizacji routingu. Pozwala ona przeliczać parametry energooszczędnej konfiguracji sieci w pętli zamkniętej z niewielkim okresem próbkowania i bez naruszania narzuconego reżimu czasowego. Po drugie, zaprezentowałem projekt pętli sterowania adaptacyjnego, która pozwala na dynamiczne dostosowanie polityki sterowania przydziału usług w węzłach sieci do zmiennych warunków operacyjnych. Po trzecie, zaprojektowałem energooszczędne sterowniki CPU bazujące na rozwiązaniu problemu sterowania optymalnego na horyzoncie nieskończonym.

Rozważmy klaster N połączonych ze sobą serwerów. Węzeł sieci, serwer n, gdzie $n = 1,...,N$, można opisać wskaźnikiem efektywności Q_n określającym koszt energetyczny przetwarzania pakietów sieciowych lub zadań obliczeniowych. Wektor incydencji (łącze-węzeł), $\mathbf{b}_n = [b_{en}]_{E\times 1}$, można w tym modelu wykorzystać do określenia sumarycznego strumienia pakietów trafiającego do serwera. Jeżeli zatem istnieje łącze przyległe do węzła n, to $b_{en} = 1$, a w przeciwnym



Rysunek 2: Architektura zaprojektowanego systemu sterownia siecią.

wypadku $b_{en} = 0$. Obciążenie serwera strumieniem pakietów napływających z zadanym natężeniem można wyrazić jako $\mathbf{b}_n^T\mathbf{r}$, gdzie $\mathbf{r} = \mathbf{Ax}$. Zauważmy, że $\mathbf{B} = [\mathbf{b}_1|...|\mathbf{b}_N]$ jest macierzą incydencji sieci.

Przypuśćmy, że dysponujemy prognozami natężenia ruchu na ścieżkach routingu, $\mathbf{d} = \hat{\mathbf{d}}(t|\boldsymbol{\beta})$, gdzie t oznacza chwilę próbkowania a $\boldsymbol{\beta}$ wektor parametrów predykcji. Następujące zadanie pozwala **wyznaczyć optymalna konfigurację routingu** w sieci oraz poziomy aktywności jej węzłów:

$$\texttt{SYSTEM}[(Q_1,...,Q_N),\mathbf{A},\mathbf{B},\mathbf{c},\mathbf{d},\mathbf{H}]:$$

$$\begin{array}{ll} \text{zminimalizuj} & \sum_{n=1}^{N} Q_n(\mathbf{b}_n^T\mathbf{Ax}) \\ \text{przy ograniczeniach} & \mathbf{Ax} - \mathbf{c} \le 0, \\ & \mathbf{Hx} - \mathbf{d} \ge 0,\ \mathbf{x} \ge 0. \end{array}$$

W powyższym sformułowaniu $\mathbf{A}$ oznacza macierz routingu, $\mathbf{H}$ macierz źródło-ścieżka, a wektor $\mathbf{x}$ rozkład przepływów źródłowych pomiędzy ścieżkami w sieci.

Ponieważ funkcja Q_n modelująca koszt energetyczny przetwarzania pakietów nie jest zazwyczaj wypukła, zadanie to należy rozwiązywać odpowiednimi technikami, np. uwypukla-

jącymi [5, 10, N4]. Jego rozwiązanie pozwala jednak wyznaczyć przydział ścieżek do przepływów optymalizując przy tym zużycie zasobów systemowych. Ponadto, ponieważ $\hat{\mathbf{r}} = \mathbf{A}\hat{\mathbf{x}}$, rozwiązanie zadania pozwala **wyznaczyć efektywne ścieżki routingu** odpowiadające obserwowanemu ruchowi sieciowemu. Łącza, dla których $\hat{r}_e = 0$ wskazują węzły znajdujące się w stanie oczekiwania, mogą zatem zostać uśpione. Zaprojektowana na podstawie powyższego sformułowania problemu struktura systemu sterowania pozwala zatem koordynować pracę sterowników optymalizujących prace serwerów (procesorów).

W projekcie efektywnych energetycznie mechanizmów sterowania procesorami uwzględniłem wyniki moich badań w obszarze sterowania optymalnego [N2]. Rozważmy wskaźnik jakości łączący koszt energetyczny pracy procesora $Q_p(y,v)$ z kosztem utraty jakości $Q_q(y,v)$ realizowanych procesów ma postać

$$Q_\gamma(y,v) = \gamma Q_p(y,v) + (1-\gamma)Q_q(y,v), \tag{10}$$

gdzie y jest obciążeniem CPU a v częstotliwością (wydajnością obliczeniową) jego pracy. Parametr γ określa stopień energooszczędności polityki sterowania. Dysponując modelem dynamiki pracy serwera $\mathcal{M}(\gamma,\boldsymbol{\theta},\boldsymbol{\vartheta}) = (Q_\gamma, f_{\bar{\boldsymbol{\theta}}}, h_{\bar{\boldsymbol{\vartheta}}})$, można sformułować następujące zadanie syntezy polityki sterowania:

$\texttt{SERVER}[Q_\gamma, f_{\bar{\boldsymbol{\theta}}}, h_{\bar{\boldsymbol{\vartheta}}}]$:

$$\left|\begin{array}{ll} \text{zminimalizuj} & J_\pi(z_0) = \lim\limits_{N\to\infty} \mathrm{E}\{\sum\limits_{k=0}^{N-1} \beta^k Q_\gamma(y_k,v_k) \colon w_k \in \mathbb{W},\ w_k \sim \mathbf{P}_k\} \\ \text{przy ograniczeniach} & z_{k+1} = f_{\bar{\boldsymbol{\theta}}}(z_k,v_k,w_k) \in \mathbb{Z}, \\ & y_k = h_{\bar{\boldsymbol{\vartheta}}}(z_k,v_k,w_k) \in \mathbb{Y}, \\ \text{względem} & \pi \in \{\{\mu_k\}_{k=0}^{N-1} | \mu_k \colon \mathbb{Z}\times\mathbb{W} \to \mathbb{V}\}. \end{array}\right.$$

Badając rozwiązania tak postawionego wielokryterialnego zadania sterowania optymalnego na horyzoncie nieskończonym **określiłem strukturę optymalnej polityki sterowania** [N2]. Polityka ta jest ograniczona od dołu przez funkcję najlepszej odpowiedzi serwera na chwilowe obciążenie, $\hat{\mu}_\gamma$, minimalizującą krótkookresowy koszt pracy. Jej strukturę opisuje następujące twierdzenie [N2].

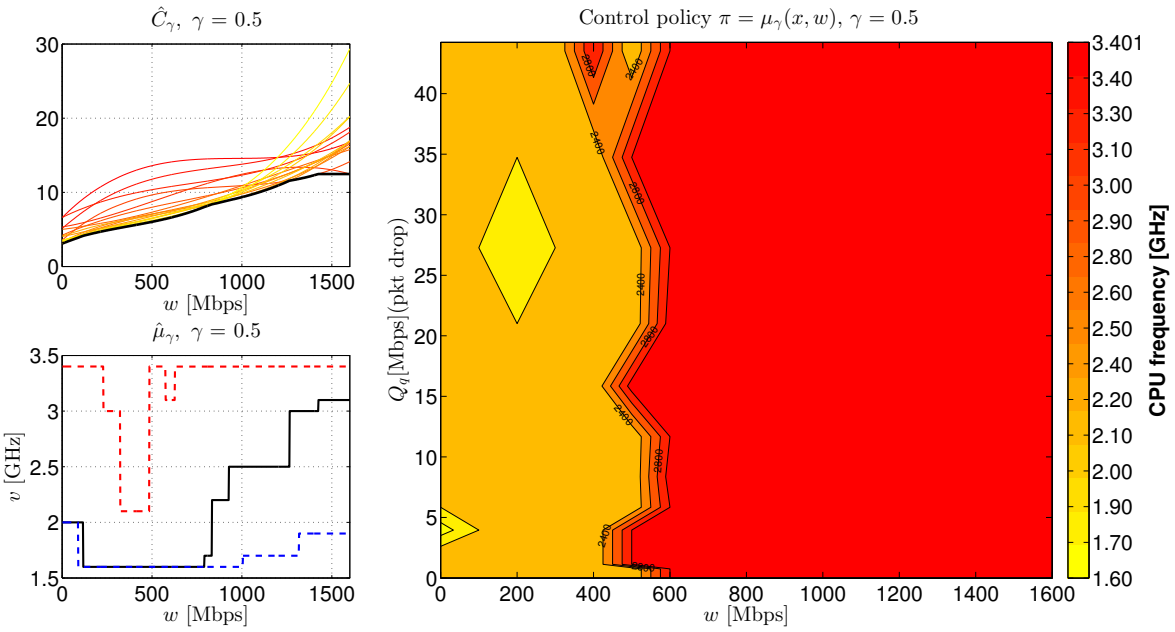
Twierdzenie 2. *Jeżeli funkcja $\boldsymbol{J}$ jest rozwiązaniem równania Bellmana dla problemu* SERVER*, to optymalna polityka sterowania ma następującą strukturę:*

$$\mu_\gamma(z,w) = \begin{cases} \hat{\mu}_\gamma(z,w), & \text{jeżeli } f_{\bar{\boldsymbol{\theta}}}(z,\hat{\mu}_\gamma(z,w),w) = 0, \\ \hat{\mu}^+_\gamma(z,w), & \text{jeżeli } f_{\bar{\boldsymbol{\theta}}}(z,\hat{\mu}_\gamma(z,w),w) > 0, \end{cases}$$

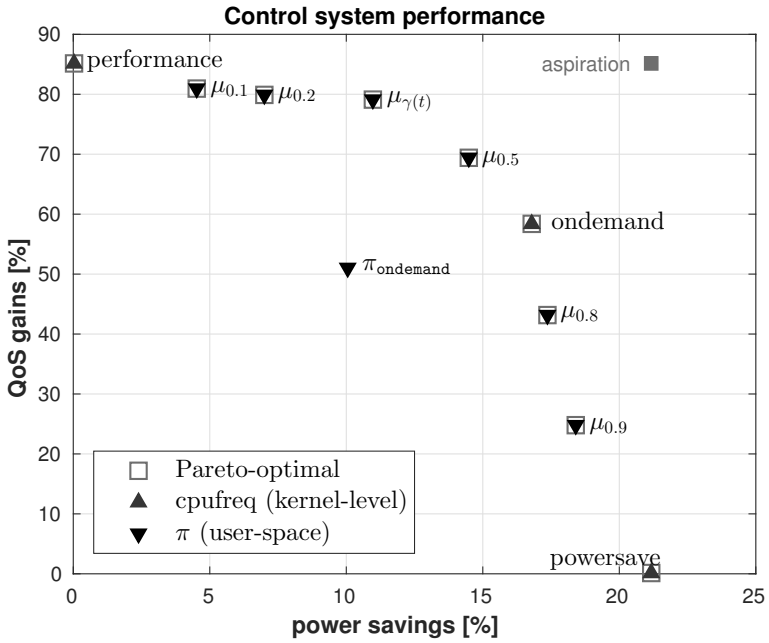
$$\hat{\mu}^+_\gamma(z,w) = \arg\min\{Q_\gamma(h_{\bar{\boldsymbol{\vartheta}}}(z,v,w),v) + \alpha J(f_{\bar{\boldsymbol{\theta}}}(z,v,w)) \colon v \geq \hat{\mu}_\gamma(z,w)\},$$

gdzie $\hat{\mu}_\gamma$ jest funkcją najlepszej odpowiedzi.

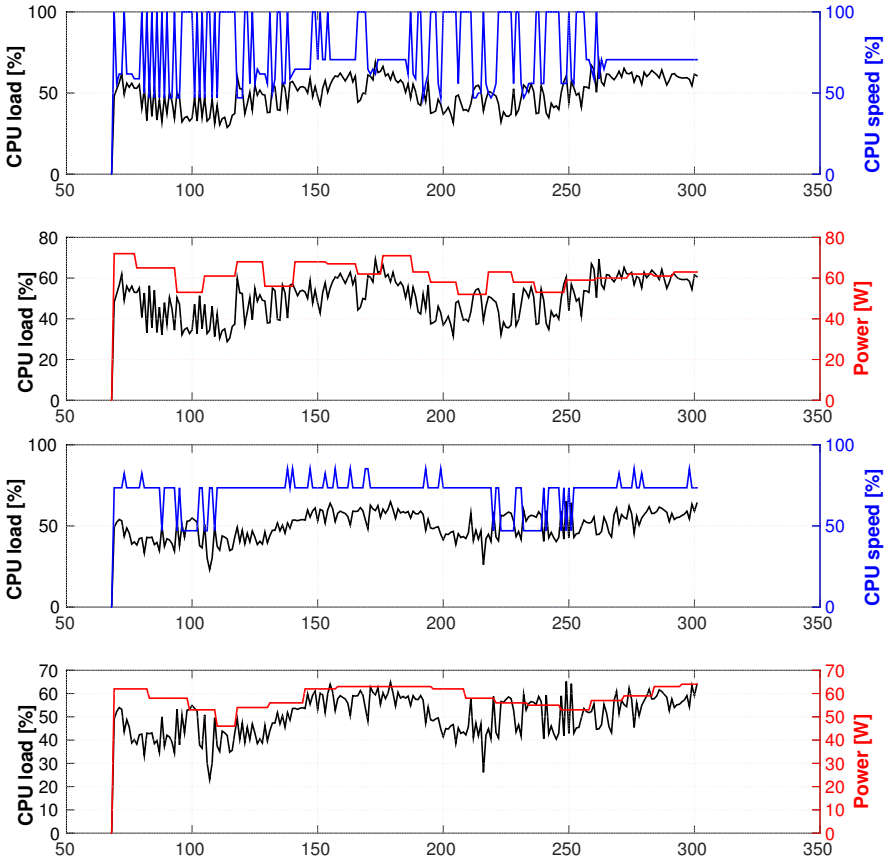
Wyznaczona polityka sterownia ma następująca interpretację. Jeśli możliwe jest obsłużenie strumienia pakietów w z minimalną częstotliwością optymalizującą koszt chwilowy, należy



Rysunek 3: Przykład polityki sterowania $\mu_{0.5}$ wyznaczonej metoda iteracji polityki sterowania na horyzoncie nieskończonym dla danych eksperymentalnych. Funkcja $\hat{C}_\gamma$ określa minimalny koszt obsługi strumienia zadań i jest wyznaczona przez obwiednie kosztów przetwarzania pakietów z dostępnymi częstotliwościami.



Rysunek 4: Prównanie wydajności sterowników CPU.

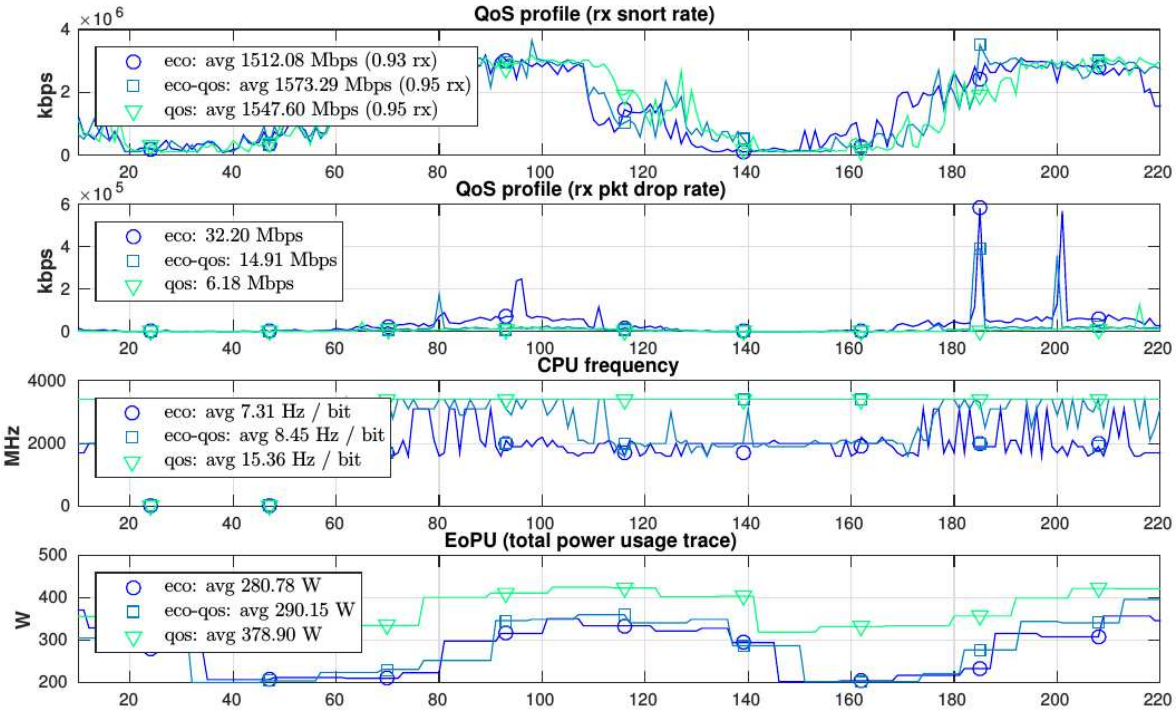


Rysunek 5: Trajektorie sterowań i pobór mocy serwera pracującego zgodnie z polityką `ondemand` (powyżej), polityki $\mu_{0.5}$ (poniżej).

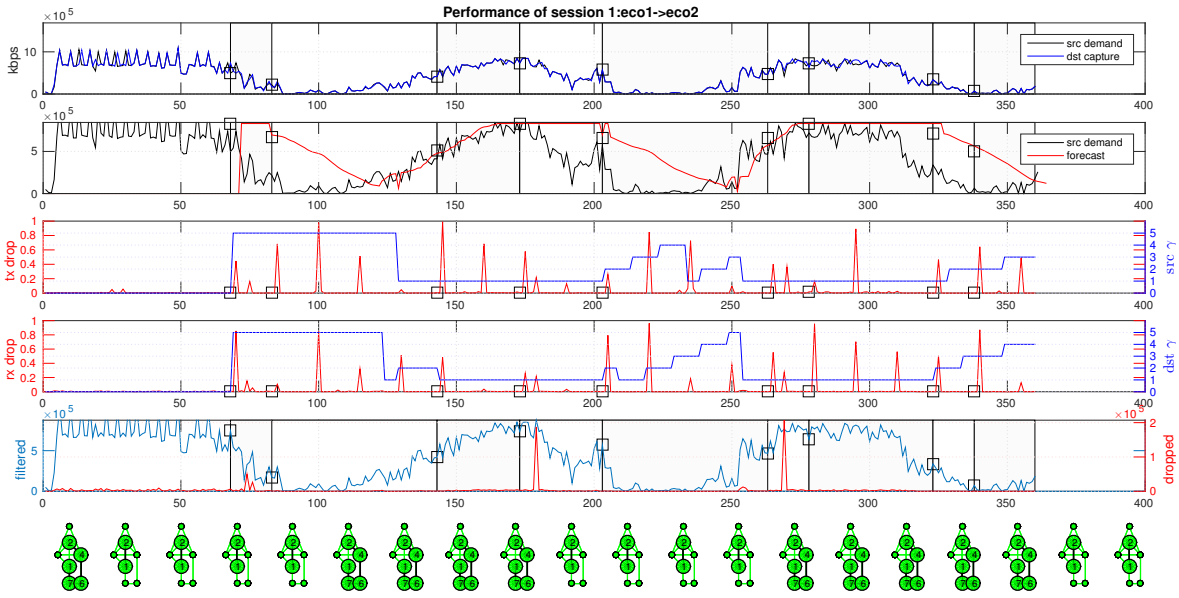
wybrać sterownie $\hat{\mu}_\gamma(z,w)$. W przeciwnym przypadku, należy optymalizować koszt długookresowy wybierając politykę ograniczoną od dołu przez $\hat{\mu}_\gamma(z,w)$.

Zaprojektowane zgodnie z opisaną wcześniej metodyką mechanizmy zostały **przebadane w laboratorium NASK PIB**. Znakomita część eksperymentów została zaprojektowana i przeprowadzona w celu kalibracji parametrów opracowywanego systemu sterowania oraz w celu estymacji wskaźnika wydajności energooszczędnego przetwarzania pakietów w śro-dowiskach systemu Linux wykorzystujących serwery odgrywające rolę zarówno routerów software'owych, jak i podlegających im użytkowników sieciowych, którzy te pakiety przetwarzają.

Zebranie reprezentatywnych danych, z których można byłoby wyciągnąć rzeczywiste wnioski na temat rozważanej technologii generowania energooszczędnego ruchu sieciowego, wymagało zastosowania dwóch typów obciążenia sieci, a mianowicie (1) zaszumione sygnały sinusoidalne o regulowanej częstotliwości i poziomie zaszumienia oraz (2) odtwarzany ruch sieciowy, wygenerowany w sposób sztuczny (www.caid.org/data). Strumienie pakietów generowane za pomocą `iperf` czy `mgen` krążyły wewnątrz opracowywanej sieci (posiadały w niej zarówno adres źródłowy jak i docelowy). Reprezentatywny przebieg eksperymentu zakładał



Rysunek 6: Porównanie polityk sterowania siecią.



Rysunek 7: Efektywność przetwarzania danych oraz proces zmian konfiguracji routingu.

transfer ruchu wychodzącego z czterech z siedmiu węzłów (eco1-7) eksperymentalnej sieci odtwarzającej topologie szkieletu sieci NASK S.A.

Pierwsza grupa eksperymentów została zaprojektowany z myślą o porównaniu efektywności jedynie lokalnej polityki sterowania (procesorem), przy założeniu stałej konfiguracji routingu. Porównane zostały następujące sterowniki CPU:

- μ_γ, $\gamma \in \{0.1, 0.2, 0.5, 0.8, 0.9\}$ (warstwy nadrzędnej)
- adaptacyjny $\mu_{\gamma(t)}$ (lokalny)
- $\pi_{\texttt{ondemand}}$, algorytm `ondemand` w przestrzeni użytkownika
- `ondemand` sterownik w przestrzeni jądra

Za rozwiązanie referencyjne przyjęto sterownik `ondemand`, podstawowy mechanizm skalowania częstotliwości CPU (sterownik `cpufreq`) dla systemu Linux. Sterownik `ondemand` dynamicznie dostosowuje częstotliwość CPU w oparciu o zaobserwowaną wartość obciążenia CPU. Za każdym razem, gdy wartość ta przekracza ustalony przez użytkownika próg, sterownik podnosi częstotliwość CPU do maksymalnej wartości akceptowalnej. Jeśli zaobserwowany poziom obciążenia spada poniżej ustalonego progu, wówczas częstotliwość przełączana jest na najniższą akceptowalną wartość, przy której zaobserwowane obciążenie będzie mogło zostać obsłużone. W eksperymentach przebadane zostały również dwa pozostałe podstawowe sterowniki, `performance` oraz `powersave`, które utrzymują częstotliwości CPU odpowiednio na najwyższym lub najniższym poziomie.

Rysunek 4 ilustruje średnią wydajność polityki lokalnej w sieci o stałej konfiguracji routingu. Sterowniki μ_γ dawały wyniki Pareto-optymalne, przewyższając $\pi_{\texttt{ondemand}}$ zarówno pod kątem oszczędności energii, jak i efektywności obliczeniowej. Szczególnie w porównaniu z $\pi_{\texttt{ondemand}}$:

- sterownik adaptacyjny $\mu_{\gamma(t)}$ **zwiększał wydajność obliczeniową** o 30% utrzymując pobór mocy na podobnym poziomie,
- sterownik $\mu_{0.5}$ **zwiększał wydajność o 20% redukując jednocześnie pobór mocy** o 5%.

Ekstremalne wartości γ zwracały wyniki podobne do tych wygenerowanych przez uruchomienie polityk `performance` oraz `powersave`. Takie zachowanie dokładnie pokrywa się z oczekiwaniami wobec zaproponowanej koncepcji lokalnej polityki sterowania.

Proces sterowania CPU realizowany w przestrzeni użytkownika może generować znaczne obciążenie CPU. Dlatego też obserwacji poddano również wyniki zwracane przez implementację sterownika `ondemand` w jądrze, które zazwyczaj mieściły się we froncie Parety (północnowschodnia granica powłoki wypukłej zbioru) wyników generowanych przez μ_γ. Rezultat ten **potwierdza hipotezę o dominacji polityki μ_γ zaimplementowanej na poziomie jądra na polityką** `ondemand`. Rysunek 5 przedstawia trajektorie wejść i wyjść sterowania mierzone dla dwóch typów sterowników CPU.

Drugi zbiór eksperymentów został zaprojektowany wyłącznie z myślą o przebadaniu wydajności energooszczędnego sterowania konfiguracją routingu. Rysunek 6 ilustruje średnią wydajność porównywanych sterowników. W obu przypadkach system sterowania dostosowywał konfigurację routingu oraz stan poboru mocy serwerów w zależności od predykcji natężenia ruchu. Lokalnie zostały zastosowane różne polityki sterowania LCP, tzn. $\mu_{0.75}$ (`eco`), $\mu_{gamma(t)}$ (`eco-qos`) oraz $\mu_{0.1}$ (`qos`). Przedmiotem obserwacji były poniższe metryki:

- szybkość przetwarzania pakietów,
- ilość odrzuconych pakietów
- ilość cykli CPU na bit (odfiltrowanego ruchu sieciowego)
- momentalny pobór mocy.

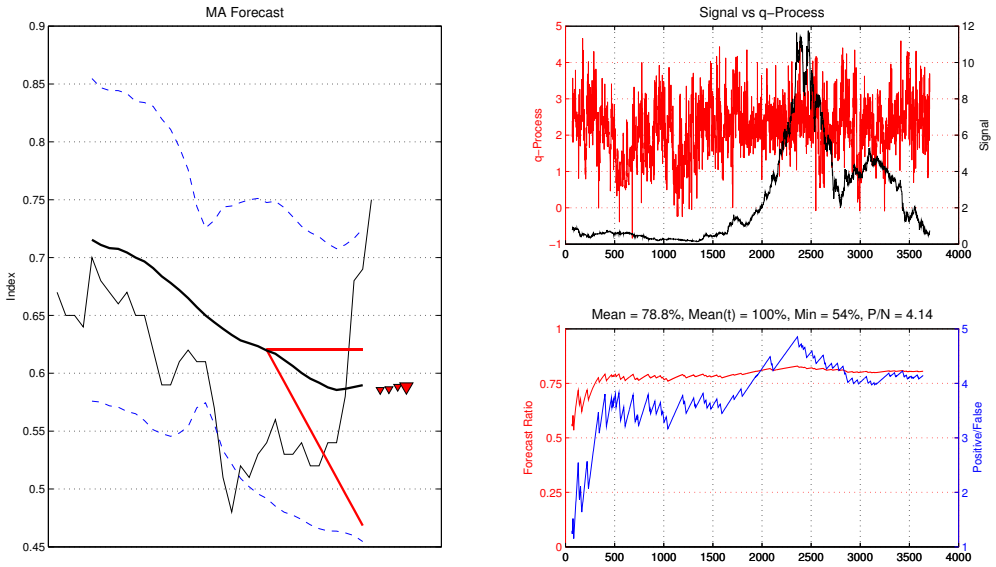
W przeprowadzonych badaniach najwyższą jakość usług uzyskał sterownik `qos`, który zdołał utrzymać maksymalną częstotliwość CPU generując w ten sposób najniższy wskaźnik odrzuconych pakietów. Działo się to oczywiście kosztem poboru mocy. Jednak stosując adaptacyjną strategię `eco-qos` udało się zredukować pobór mocy do poziomu porównywalnego z tym osiąganym przez energooszczędnę politykę `eco`. Jednocześnie jakość usług różniła się od jakości osiąganej przez sterownik `qos`, aczkolwiek najczęściej w okresach wysokiej wykorzystania sieci. Szczegółowa analiza zebranych danych pokazuje, że nadrzędne polityki `qos` oraz `eco-qos` zapewniały zbliżoną średnią jakość usług.

Rysunek 7 prezentuje przykładowe trajektorie sterowania konfiguracją sieci. Pierwszy wykres pokazuje zgodność ilości pakietów wysłanych i odebranych w węzłach źródłowych i docelowych. Drugi wykres dostarcza dodatkowych informacji o prognozowanych wartościach szczytowego natężenia ruchu w sesji, które stanowią podstawę decyzji o routingu. Ścieżki routingu przypisane do sesji w zależności od otrzymanej prognozy, zostały zaprezentowane na środkowym wykresie. Jak można zauważyć, polityka sterowania routingiem przekierowuje ruch sesji do alternatywnej (dłuższej) ścieżki, tym samym odciążając węzły pierwotnie wysycone w okresie wyższego zapotrzebowania na ogólną przepustowość sieci. Taka konfiguracja tablic routingu była podyktowana wymogiem utrzymania jakości wszystkich sterowanych sesji na żądanym poziomie. Trzy pozostałe wykresy pokazują relację pomiędzy ilością odrzuconych pakietów, konfiguracją lokalnej energooszczędnej polityki procesora oraz całkowitą ilością pakietów przefiltrowanych.

W trakcie prac nad zagadnieniem prognozowania obciążenia sieci wykorzystałem wyniki swoich badań nad **syntezą algorytmów prognozowania szeregów czasowych** wykorzystujących następujące niestacjonarne równanie (dyfuzji) ułamkowego rzędu:

$$\left[\frac{\partial}{\partial x^2} + \sigma^{p(\tau)}\frac{\partial^{p(\tau)}}{\partial t^{p(\tau)}}\right] u(x,t) = F(x,t) \tag{11}$$

Prace te, wciąż znajdujące się we wczesnym stadium, zaowocowały projektem nowego algorytmu predykcji wykorzystującego ułamkowy i niestacjonarny model procesu dyfuzji, umożliwiający skuteczne prognozowanie szeregów czasowych o zmiennej charakterystyce.



Rysunek 8: Prognozowania zmian trendu kursu giełdowego.

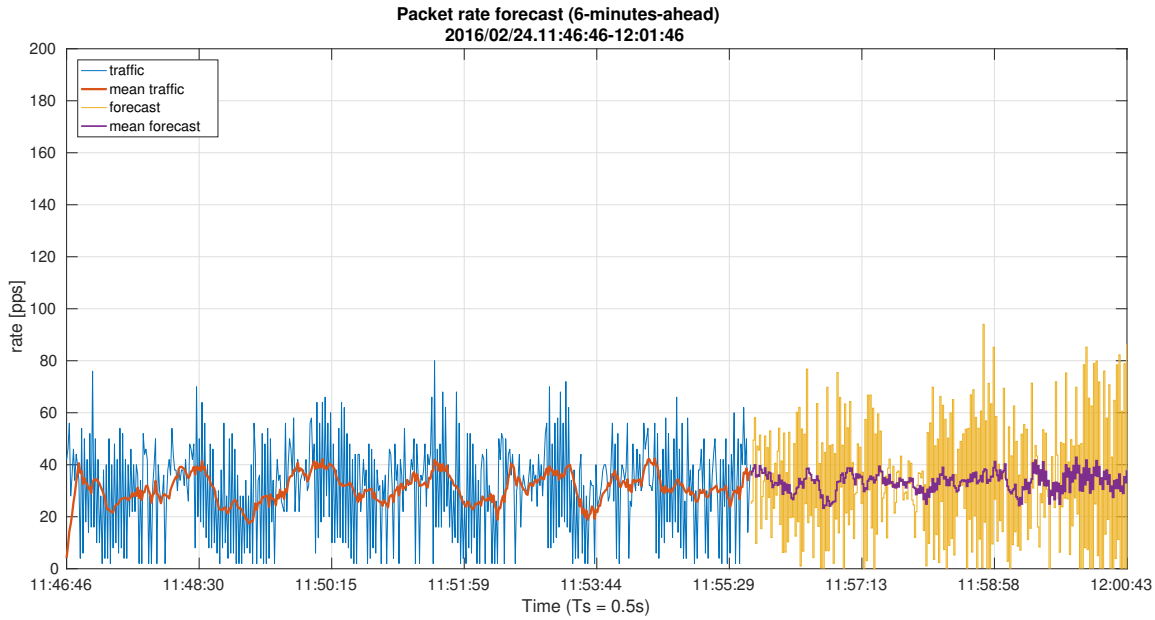
Opracowany przeze mnie na podstawie pracy [4] algorytm **oblicza krótkookresową prognozę wzrostów lub spadków natężenia ruchu** na podstawie wyników testu hipotezy opisującej zależność pomiędzy dynamiką ułamkowego rzędu $p(\tau)$ równania dyfuzji, identyfikowanego w zadanym oknie czasowym τ, a wartością oczekiwaną prognozowanego procesu stochastycznego. Hipoteza sugeruje, że zwiastunem zmiany trendu procesu jest zmiana modelu dynamiki wyrażonego ułamkowym rzędem równania. Rysunek 8 przedstawia wyniki prognozowania wzrostów i spadków wartości niestacjonarnego szeregu czasowego (kursu giełdowego), przy **prawdopodobieństwie trafności prognozy wynoszącym ponad 75%**.

Ze względu na charakter obciążenia łączy sieciowych, jak wiadomo są to procesy stochastyczne często przejawiające cechy samopodobieństwa lub ciężko-ogonowe [12, 20], opracowany przeze mnie algorytm znalazł **zastosowanie w aplikacji prognozujące nagłe zmiany obciążenia sieci** akademickiej zarządzanej przez NASK PIB. Rozwiązanie to znalazło również bezpośrednie zastosowanie w systemie `FLDX`.

3.3 Synteza mechanizmów sterowania adaptacyjnego

Doświadczenia zgromadzone podczas prac nad syntezą algorytmów sterowania optymalnego metodą programowania dynamicznego poszerzyłem prowadząc badania nad syntezą adaptacyjnych algorytmów sterowania procesami o zmiennej dynamice. Na podstawie wyników badań uzyskanych przeze mnie w tym obszarze **zaprojektowałem oraz zaimplementowałem adaptacyjne mechanizmy sterowania**, które zostały wykorzystane:

- w systemie `FLDX` do tłumienia ataków DDoS oraz kształtowania dynamiki przepływów



Rysunek 9: Ilustracja działania algorytmu prognozowania natężenia ruchu na łączu sieci akademickiej NASK. Prognoza wystawiana w czasie rzeczywistym na podstawie obserwowanego strumienia NetFlow.

monitorowanych,

- w jądrze systemu Linux do energooszczędnego sterowania częstotliwością pracy procesora CPU.

W badaniach wykorzystałem równania syntezy wyprowadzane dla modelu wielomianowego:

$$\begin{aligned} A(q)x(t) &= B(q)(u(t)+v(t)),\\ y(t) &= x(t)+e(t),\\ R(q)u(t) &= T(q)u_c(t)-S(q)y(t), \end{aligned} \tag{12}$$

gdzie q oznacza operator przesunięcia, $u(t)$ jest sygnałem sterowania, $v(t)$ zaburzeniem, $x(t)$ wew-nętrznym stanem procesu, $e(t)$ błędem pomiaru obserwowanego wyjścia $y(t)$. Dynamika stero-wanego procesu określona jest w rozważanym modelu przez parę wielomianów:

$$\begin{aligned} A &\triangleq A(q) = q^n + a_1 q^{n-1} + \cdots + a_n,\\ B &\triangleq B(q) = b_0 q^{n_b} + b_1 q^{n_b-1} + \cdots + b_{n_b},\\ d &= \deg A - \deg B = n - n_b > 0. \end{aligned} \tag{13}$$

Przyjąłem przy tym oczywiście założenie, że model dynamiki (A,B) jest nieznany i zmienny w czasie. Założenie to wymusza identyfikację modelu dynamiki w czasie realizacji procesu sterowania oraz nakazuje uwzględnienie w procesie sterowania obecności nieusuwalnych niedokładności modelu dynamiki.

Zadanie syntezy polega na obliczeniu algorytm sterowania postaci:

$$R(q)u(t) = T(q)u_c(t) - S(q)y(t). \tag{14}$$

określonego przez wielomiany:

$$\begin{aligned} R &\triangleq R(q) = r_0 q^{n_r} + r_1 q^{n_r-1} + \cdots + r_{n_r}, \\ T &\triangleq T(q) = t_0 q^{n_t} + t_1 q^{n_t-1} + \cdots + t_{n_t}, \\ S &\triangleq S(q) = s_0 q^{n_s} + s_1 q^{n_s-1} + \cdots + s_{n_s}. \end{aligned} \tag{15}$$

Wymagane jest tutaj zachowanie warunków przyczynowości:

$$\begin{aligned} &\deg R \geq \deg S, \\ &\deg R \geq \deg T. \end{aligned} \tag{16}$$

Celem sterowania realizowanego przez poszukiwany algorytm jest kształtowanie przebiegu sygnału $x(t)$ w sposób minimalizujący wariancję uchybu regulacji:

$$\varepsilon(t) = u_c(t) - x(t) \tag{17}$$

przy ograniczeniach obejmujących charakterystyki tłumienia wejść swobodnych $v(t)$ i $e(t)$ w warunkach niepewności wynikających z nieznajomości dokładnego modelu dynamiki procesu sterowania.

Zgodnie z przyjętymi założeniami przebieg procesu sterowania będzie modelowany przez układ równań:

$$\begin{bmatrix} 0 & S & R \\ A & 0 & -B \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \\ u(t) \end{bmatrix} = \begin{bmatrix} Tu_c(t) \\ Bv(t) \\ e(t) \end{bmatrix}. \tag{18}$$

Dynamika systemu określona jest wobec powyższego przez wielomian charakterystyczny postaci:

$$\det\left(\begin{bmatrix} 0 & S & R \\ A & 0 & -B \\ -1 & 1 & 0 \end{bmatrix} \right) = AR + BS. \tag{19}$$

Relacje opisujące wpływ sygnałów wejściowych układu, (u_c, v, e), na przebieg sygnałów wyjściowych, (x, y, u), mają wobec przyjętych założeń klasyczną postać określoną przez rozwiązanie układu równań (18):

$$\begin{bmatrix} x(t) \\ y(t) \\ u(t) \end{bmatrix} = H \begin{bmatrix} u_c(t) \\ v(t) \\ e(t) \end{bmatrix}, \text{ gdzie } H = \begin{bmatrix} \frac{BT}{AR+BS} & \frac{BR}{AR+BS} & \frac{-BS}{AR+BS} \\ \frac{BT}{AR+BS} & \frac{BR}{AR+BS} & \frac{AR}{AR+BS} \\ \frac{AT}{AR+BS} & \frac{-BS}{AR+BS} & \frac{-AS}{AR+BS} \end{bmatrix}. \tag{20}$$

Synteza algorytmu sterowania sprowadza się do wyznaczenia współczynników wielomianów (R,T,S) kształtujących sygnały wyjściowe według przyjętych wymagań projektowych. Wymagania te określone są przez parę wielomianów:

$$\begin{aligned} A_m &\triangleq A_m(q) = a_{m0}q^{n_a} + a_{m1}q^{n_a-1} + \cdots + a_{mn_a}, \\ B_m &\triangleq B_m(q) = b_{m0}q^{n_{b_m}} + b_{m1}q^{n_{b_m}-1} + \cdots + b_{mn_{b_m}}, \\ d_m &= \deg A_m - \deg B_m \geq \deg A - \deg B, \end{aligned} \tag{21}$$

oraz przez specyfikację dotyczącą charakterystyk amplitudowo-fazowych $H(\omega)$. Problem ten jest równoznaczny z problemem lokowania biegunów i sprowadza się do wyznaczenia rozwiązania równania [2, 3]:

$$\frac{BT}{AR+BS} = \frac{B_m}{A_m}. \tag{22}$$

przy ograniczeniach zadanych wymaganiami filtracji, stabilności i odporności układu sterowania. Przyjęte zostało dodatkowo założenie dokładnego śledzenia modelu, $\deg A = \deg A_m = n$.

Algorytm lokowania biegunów

W przyjętym podejściu synteza algorytmu sterowania prowadzona jest zgodnie z klasyczną metodyką lokowania biegunów. Dokonajmy rozkładu wielomianu B na wielomiany B^+ oraz B^-:

$$B = B^+B^-. \tag{23}$$

Wobec zer wielomianu B^+ stawiane jest następujące wymaganie:

$$\bar{z}^+ \in \Upsilon(B^+) \triangleq \{z \in \mathbb{C} | B^+(z) = 0 \wedge |z| < \gamma \ll 1\}. \tag{24}$$

Zera spełniające powyższy warunek mogą być kasowane przez regulator bez ryzyka utraty stabilności układu. Przyjęte zostało również założenie unormowania współczynników wielomianu, tzn. $b_0^+ = 1$. Rozważamy równanie syntezy:

$$\frac{BT}{AR+BS} = \frac{B_m}{A_m}. \tag{25}$$

Poszukiwany regulator (R,T,S) powinien przekształcać układ (A,B) do postaci (A_m,B_m). Przekształcenie to można uzyskać przeprowadzając następującą serię faktoryzacji:

$$\frac{BT}{AR+BS} = \frac{B^+B^-T}{AB^+\bar{R}+B^+B^-S} = \frac{B^-T}{A\bar{R}+B^-S} = \frac{B_m}{A_m}. \tag{26}$$

Należy podkreślić, że powyższa faktoryzacja jest dopuszczalna jedynie w przypadku skracania wielomianów B^+ posiadających zera $\bar{z} \in \Upsilon(B^+)$. Przyjmując powyższą zasadę równanie syntezy można również zapisać w następujący uogólniony sposób:

$$\frac{B^-T}{A\bar{R}+B^-S} = \frac{A_oB_m}{A_oA_m}, \tag{27}$$

gdzie A_o jest stabilnym wielomianem obserwatora wyjścia posiadającym zera $\bar{z} \in \Upsilon(A_o)$. Wprowadzenie do równania dodatkowego wielomianu obserwatora, A_o, pozwala uwzględnić w syntezie ograniczenia zadane wymaganiami filtracji, stabilności i odporności układu.

Algorytm lokowania biegunów sprowadza się do rozwiązania równania diofantycznego:

$$AR_d\bar{R} + B^- S_d \bar{S} = A_f A_o A_m \tag{28}$$

względem wielomianów $\bar{R}$ oraz $\bar{S}$, gdzie R_d, S_d i A_f określają dodatkowe parametry specyfikacji procesu sterowania. Wielomian T ma tutaj postać:

$$T = \rho A_o \bar{B}_m. \tag{29}$$

Współczynnik ρ określa wzmocnienie sygnału wartości zadanej, $u_c(t)$, w stanie ustalonym. Jego wartość można wyznaczyć korzystając z twierdzenia o wartości końcowej. Załóżmy, że $u_c(t) = \mathbb{1}(t)$ oraz $v(t) = 0$ i $e(t) = 0$. Wówczas:

$$\begin{aligned} U_c(z) &= (1 - z^{-1})^{-1}, \\ X(z) &= \frac{B(z)T(z)}{A(z)R(z) + B(z)S(z)} U_c(z). \end{aligned} \tag{30}$$

Na mocy twierdzenie o wartości końcowej:

$$\lim_{t\to\infty} x(t) = \lim_{z\to 1}(1 - z^{-1})X(z) = \frac{B(1)T(1)}{A(1)R(1) + B(1)S(1)} = \rho \frac{B(1)A_o(1)\bar{B}_m(1)}{A(1)R(1) + B(1)S(1)} = \rho H_{11}(1). \tag{31}$$

Dla zadanej wartości wyjścia w stanie ustalonym, $x^* = \lim_{t\to\infty} x(t)$, otrzymujemy współczynnik wzmocnienia $\rho = x^*/H_{11}(1)$.

Poszukiwany algorytm sterowania ma postać określoną przez następujące wielomiany:

$$\begin{aligned} R &= B^+ R_d \bar{R}, \\ T &= \rho A_o \bar{B}_m, \\ S &= S_d \bar{S}. \end{aligned} \tag{32}$$

Opisana powyżej technika syntezy została wykorzystana przeze mnie **w projekcie i implementacji kolejnej wersji sterownika CPU, a także w mechanizmach tłumienia ataków DDoS systemu** `FLDX`. Implementacja powyższej metody syntezy w środowisku działajacego operacyjnie systemu sterowania zaliczam do swoich głównych osiągnięć w tym obszarze. Zadanie to jest złożone i wymaga rozwiązania wielu problemów algorytmicznych oraz technicznych. Do głównych osiągnięć badawczych w tym obszarze zaliczam:

- implementację algorytmu syntezy regulatora (R, T, S) rozwiązującego równanie diofantyczne,
- identyfikację współczynników filtru S_d wygładzającego sygnał sprzężenia zwrotnego oraz
- identyfikację adekwatnego modelu referencyjnego (A_m, B_m).

Filtry te realizują postawione w procesie projektowania układu sterowania wymagania dotyczące kształtowania przepływów sieciowych oraz obciążenia procesora. Badania towarzyszące tym pracom, opisane przeze mnie w pracach [1, N7, 7], **poszerzają również wiedzę na temat dynamiki zjawisk związanych z przetwarzaniem danych na poziomie kart sieciowych, buforów pamięci procesorów oraz kolejek urządzeń sieciowych**. Podejście to umożliwia także skuteczne korygowanie pracy mechanizmów policingu przełączników sieciowych.

4 Struktura i opis praktycznej części zasadniczego osiągnięcia naukowego

Wykorzystanie wyników moich prac badawczych obejmuje osiągnięcie wchodzące w skład dzieła głównego oraz wdrożenia towarzyszące i dodatkowe.

1. Główne osiągnięcie: opracowanie projektu architektury, implementacja i wdrożenie systemu `FLDX`.
2. Osiągnięcie towarzyszące: opracowanie i implementacja rodziny energooszczędnych, adaptacyjnych sterowników procesora CPU dla jądra systemu operacyjnego Linux.
3. Osiągnięcie dodatkowe: opracowanie projektu architektury, implementacja i wdrożenie systemu `ARAKIS`.

Wyniki moich badań znalazły bezpośrednie zastosowanie w zaprojektowanych przeze mnie oraz wdrożonych pod moim kierownictwem teleinformatycznych systemach cyberbezpieczeństwa, obejmujących swoim działaniem znaczną część cyberprzestrzeni Polski, w systemach `FLDX` oraz `ARAKIS`. Podejście wykorzystujące zaawansowaną matematykę teorii sygnałów i sterowania w projektowaniu architektury informatycznego systemu cyberbezpieczeństwa nie było dotychczas stosowane w rozwiązaniach dostępnych na rynku, co uzasadniło objęcie konstrukcji systemu `FLDX` ochroną patentową. Ponadto, wyniki moich prac zostały wykorzystane w projekcie i implementacji energooszczędnego sterownika procesora CPU dla jądra systemu Linux [N8].

4.1 `FLDX`: system wczesnego wykrywania i tłumienia ataków typu DDoS

Zdecydowana większość ataków DDoS statystycznie jest przeprowadzana w okresie nie dłuższym niż 30 minut, przy czym wśród nich dominują ataki trwające poniżej pięciu minut. Wyniki te wyraźnie określają aktualne wymagania dotyczące czasu reakcji, którym sprostać muszą systemy detekcji ataków oraz zespoły reagowania na incydenty naruszenia bezpieczeństwa teleinformatycznego. Obiektem ataków pozostają najważniejsze i najpopularniejsze typy usług sieciowych: `telnet`, `http`, `https`, `dns` oraz usługi serwerów gier. Techniki prowadzenia ataków bazują na stabilnym zbiorze protokołów oraz podstawowych własnościach powszechnie dostępnych usług sieciowych. Wśród nich należy wymienić, po pierwsze, naturalnie występującą asymetrię pomiędzy rozmiarem zapytania kierowanego do serwera a rozmiarem udzielanej przez serwer odpowiedzi, a po drugie, naturalnie ograniczoną wydajność mechanizmów obsługi protokołów stanowych, szczególnie TCP.

Wykorzystanie podstawowych i ogólnodostępnych mechanizmów sieciowych umożliwia efektywne konstruowanie wielu urozmaiconych wariantów ataków DDoS. Z tego względu prognozowany jest wzrost liczby ataków wielowymiarowych o złożonej dynamice, wykorzystujących jako bazę jednocześnie wiele podatności, i skierowanych przeciwko wielu celom.

Podstawowym narzędziem realizacji ataków DDoS pozostają sieci botów, maszyn zainfekowanych oprogramowaniem przygotowanym do przeprowadzenia skoordynowanego ataku DDoS według określonego scenariusza ataku. Możliwości botnetów powiększają się na wielu polach, wyścig zbrojeń w tym obszarze nie zwalnia. W świetle przedstawionych danych należy oczekiwać wzrostu stopnia zaawansowania algorytmów wykorzystywanych przez botnety do maskowania obecności, koordynacji oraz realizacji ataków.

Analiza najnowszych raportów opisujących krajobraz cyberbezpieczeństwa prowadzi wobec powyższego do następujących wniosków:

- ataki DDoS stanowią dominujący typ zagrożeń,
- rośnie skala ataków wolumentrycznych,
- należy oczekiwać dominacji ataków wielowymiarowych o złożonej i zmiennej w czasie dynamice.

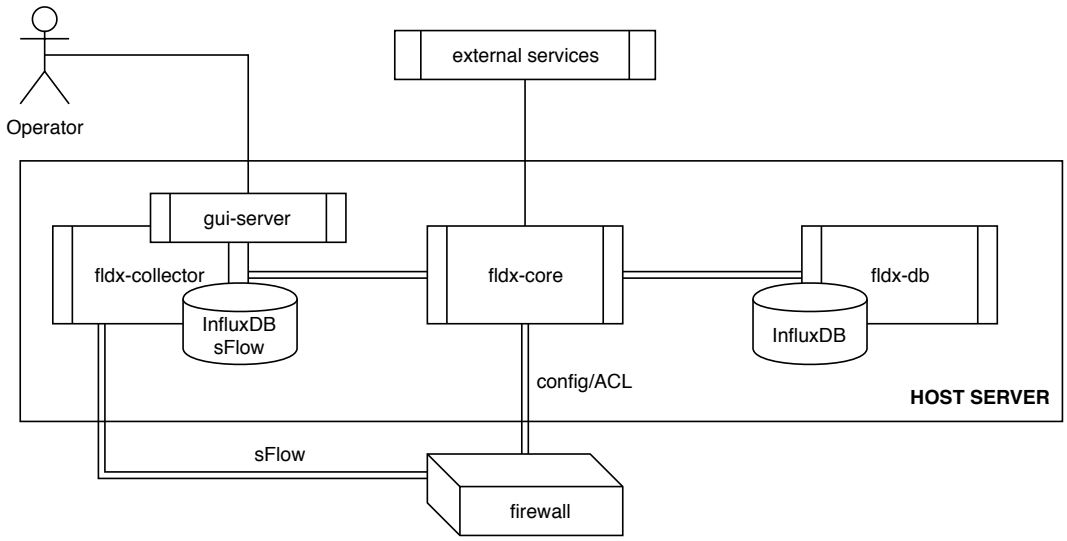
Przedstawione analizy doprowadziły mnie do sformułowania listy technologicznych wyzwań, z którymi muszą zmierzyć się projekty nowej generacji systemów ochrony sieci przed atakami DDoS, mianowicie:

- skuteczny system detekcji i tłumienia ataków DDoS musi być zdolny do szybkiego śledzenia stale uciekającego wektora ataków i podejmowania działań antycypujących scenariusze ataku w sposób bezpieczny i wiarygodny,
- system musi wzmacniać zdolność operatora sieci do reagowania w okresie kilkunastu sekund, m.in. poprzez przygotowanie skutecznych środków oddziaływania na atak, oraz udostępniać odpowiednio odfiltrowany zestaw danych (ukrywający techniczną złożoność systemu),
- system musi podnosić bezpieczeństwo i niezawodność chronionego systemu.

Sprostanie tym wyzwaniom i rozwiązanie towarzyszących im problemów technologicznych jest niezwykle trudnym i ambitnym zadaniem, zadaniem polegającym na wytworzeniu nowej technologii konkurencyjnej względem rozwiązań dostępnych na rynku. Na podstawie wyników wieloletnich badań naukowych, bazując na zaleceniach ekspertów posiadających wieloletnie i bogate doświadczenia praktyczne w obszarze cyberbezpieczeństwa, a także biorąc pod uwagę wyniki analiz rynku, opracowałem koncepcję systemu detekcji i tłumienia ataków DDoS, która odpowiada na szereg wymienionych wyżej wyzwań.

Szkic architektury systemu

Architektura systemu `FLDX` koncepcyjnie odpowiada architekturze warstowego systemu sterowania wyposażonego w pętlę sterowania nadrzędnego (nadzorczego) realizującą zadania kształtowania przepływów pakietów oraz zadania detekcji i diagnostyki anomalii występujących w monitorowanym przez system ruchu sieciowym. Bieżące wdrożenie systemu nadzoruje



Rysunek 10: Schemat blokowy architektury systemu (widok interakcji komponentów).

proces przesyłania pakietów przez przełączniki QFX Juniper warstw L2-L4 monitorując oraz kształtując przepływy przy pomocy usług udostępnianych przez system operacyjny Junos. Usługi przetwarzania danych oraz usługi obliczeniowe zostały zaimplementowane w środowisku wirtualnego klastra maszyn pracujących pod systemem operacyjnym Linux. Wytwarzane przez system dane, opisujące wykryte anomalie oraz wzorce ruchu archiwizowane są w bazie danych InfluxDB. Szeregi czasowe wytwarzane w procesie monitorowania przepływów archiwizowane są w bazie danych InfluxDB. Bazy te mogą być wykorzystane przez dowolne narzędzia wizualizacji i monitorowania danych stanowiących warstwę prezentacji systemu. System wyposażony został w zaawansowane mechanizmy administracyjne `systemd` oraz mechanizmy pamięci podręcznej optymalizujące czas dostępu do danych.

Schemat blokowy wdrażanej wersji architektury systemu `FLDX` zaprezentowany został na rysunku 10.

Architektura infrastruktury sieciowej

Transparentna architektura sieciowa L2-4 wykorzystująca powszechnie dostępne technologie sieciowe pozwala na łatwe dołączanie klientów do sieci wyposażonej w mechanizmy znoszenia ataków DDoS. Wykorzystanie zaawansowanych metod inżynierii sieciowej pozwala na realizowanie usługi systemu `FLDX` nawet na pojedynczym urządzeniu sieciowym, przekształcając przez to urządzenie typu firewall do postaci inteligentnego urządzenia ochrony sieci.

Zaprojektowana architektura infrastruktury sprzętowej umożliwia bardzo wydajne monitorowanie i filtrowanie ruchu w skali państwa. Wykorzystanie protokołu `flowspec` w sieciach operatorskich pozwala na implementację rozbudowanych i precyzyjnych reguł filtracji oraz ich zdalną instalację na urządzeniach sieciowych. Protokół `flowspec` będąc rozszerzeniem protokołu BGP nie wymusza rekonfiguracji lub zmiany architektury sieci, lecz stanowi rozwinięcie

dobrze znanych mechanizmów stosowanych przez administratorów sieci WAN.

Mechanizmy szybkiego rozpoznawania anomalii

System `FLDX` przetwarza i analizuje zbiór próbek `sFlow` pakietów przepływających przez interfejsy przełączników będących elementami infrastruktury sieciowej systemu. W obecnej konfiguracji system jest zdolny do rejestracji każdego pakietu przepływającego przez monitorowane interfejsy sieciowe (próbkowanie 1:1). Dzięki temu obserwacji podlega szerokie spektrum parametrów ruchu sieciowego oraz zjawisk o charakterze statycznym oraz dynamicznym występujących w warstwach L2-L4 modelu OSI.

Podstawowa koncepcja detekcji złośliwych zaburzeń przepływów pakietów bazuje na innowacyjnym podejściu sygnałowym. Synteza algorytmów przeprowadzona została z wykorzystaniem wyników opisanych wcześniej badań.

Zaprojektowane algorytmy detekcji izolują przepływy pakietów naruszające normy przepływów regularnych oraz porządkują je według poziomu potencjalnie stwarzanego zagrożenia. Dzięki wykorzystaniu autorskich mechanizmów wspierających, bazujących na elementach teorii zbiorów uporządkowanych oraz teorii indukcji reguł decyzyjnych, zaprojektowane mechanizmy są zdolne do bardzo wczesnego i autonomicznego wykrywania anomalii w ruchu sieciowym. System w domyślnej konfiguracji nie otrzymuje od operatora definicji przepływów chronionych, lecz samodzielnie z wysoką precyzją wykrywa przepływy wymagające monitorowania, autonomicznie dostrajając się do obserwowanej na bieżąco charakterystyki ruchu sieciowego.

Mechanizmy predykcyjnego sterowania adaptacyjnego

Dla każdego wyizolowanego przepływu pakietów w systemie `FLDX` tworzony jest obiekt regulatora, który jest odpowiedzialny za realizację procesu sterowania kształtującego dynamikę przepływu. Problem syntezy algorytmu sterowania został rozwiązany z wykorzystaniem elementów teorii sterowania adaptacyjnego, teorii procesów stochastycznych oraz teorii sygnałów. Zgodnie z przyjętym podejściem w każdej iteracji procesu sterowania adaptacyjnego przeprowadzana jest identyfikacja modelu dynamiki, a następnie dla zidentyfikowanego modelu dynamiki przeprowadzana jest synteza algorytmu sterowania oraz algorytmu predykcji. Zaprojektowane algorytmy wyznaczają kolejne wartości sygnału sterującego na podstawie prognozowanej i obserwowanej odpowiedzi źródeł ruchu sieciowego na ograniczenia szybkość przepływów pakietów. Celem sterowania jest ukształtowanie dynamiki przepływu zgodnie ze specyfikacją zadaną przez operatora systemu.

Mechanizmy translacji formuł logicznych i warstwa abstrakcji sprzętowej

Jednym z centralnych elementów systemu `FLDX` jest moduł `fldx-core` tworzący warstwę abstrakcji sprzętowej oraz udostępniający funkcje translacji formuł logicznych. Warstwa abstrakcji pośredniczy w komunikacji warstwy sterowania nadrzędnego z warstwą infrastruktury

sieciowej systemu. Odpowiada ona za odwzorowanie stanu przełączników sieciowych w warstwie sterowania oraz przekazywanie pomiędzy warstwami sygnałów sterowania. Moduł odpowiada również za przekształcanie reguł filtracji wyrażonych w języku `pcap-filter` na reguły filtracji systemu Junos.

Matematyczny model dziedziny wnioskowania

W projekcie systemu użyte zostały liczne koncepcje matematyczne oraz terminologia wywodząca się z wybranych standardów technicznych. Poniżej przedstawione zostały formalne definicje oraz słownik wykorzystywanych pojęć.

Ze względu na przyjętą koncepcję architektury systemu w jego projekcie wykorzystane zostały formalne pojęcia teorii rachunku predykatów i wnioskowania indukcyjnego (uczenia maszynowego), teorii mnogości, teorii sterowania i przetwarzania sygnałów. Podstawowymi źródłami wykorzystanych terminów i koncepcji technicznych są:

- dokumentacja systemu Junos OS (`www.juniper.net/documentation`), szczególnie w zakresie konfiguracji reguł filtracji pakietów,
- standard `sFlow` (`www.sFlow.org`) opisujący technologię próbkowania strumieni pakietów przepływających przez urządzenia sieciowe,
- dokumentacja języka filtrów pakietów `pcap-filter` (`www.tcpdump.org`) oraz filtrów `nfdump` (`github.com/phaag/nfdump`).

Zgodnie z przyjętą koncepcją system `FLDX` można traktować jako uczący się (adaptacyjny) wielowarstwowy system sterowania przepływami pakietów sieciowych odporny na zakłócenia spowo- dowane wybraną klasą zjawisk (ataków typu DDoS), wyposażony w mechanizmy diagnostyczne identyfikujące i izolujące źródła anomalii, pozyskujący (generujący) wiedzę (deklaratywną i proceduralną) na temat obserwowanych zjawisk. Poniżej zamieszczone zostały częściowo sformalizowane definicje pojęć wykorzystanych w matematycznym modelu systemu oraz jego modułów.

W tym rozdziale zaprezentowany jest opracowany przeze mnie formalny model dziedziny wnioskowania, na której przeprowadzane są wszystkie operacje systemu `FLDX`. Dla uproszczenia prezentacji podstawowych pojęć i definicji wykorzystywanych w opisie systemu w dalszej części dokumentu *pakietem* (sieciowym) nazywana będzie jednostka danych protokołu telekomunikacyjnego (PDU) modelu OSI. Wartości pól zawartych w nagłówkach protokołów oraz parametry charakteryzujące transmisję pakietu będą łącznie tworzyły dziedzinę wnioskowania, na której zaprojektowany system realizuje procesy budowania wiedzy.

dziedzina parametrów pakietów: Niech $F_i,\ i = 1,...,n$, oznacza zbiór liczb reprezentujących zawartość wybranych pól nagłówków protokołów sieciowych oraz parametry charakteryzujące pakiet sieciowy. Iloczyn kartezjański $\mathbf{F} = F_1 \times F_2 \times ... \times F_n$ będzie nazywany dziedziną parametrów pakietów (systemową dziedziną wnioskowania).

Dziedzina parametrów pakietów określona standardem `sFlow`, spełniająca powyższą definicję, może mieć następującą postać:

$$\boldsymbol{X} = \texttt{TS} \times \texttt{TE} \times \texttt{TD} \times \texttt{SA} \times \texttt{DA} \times \texttt{SP} \times \texttt{DP} \times \texttt{PR} \times \texttt{FLG} \times \texttt{FWD} \times \texttt{STOS} \times \texttt{IPKT} \times \texttt{IBYT} \tag{33}$$

$$\times \texttt{OPKT} \times \texttt{OBYT} \times \texttt{IN} \times \texttt{OUT} \times \texttt{SAS} \times \texttt{DAS} \times \texttt{SMK} \times \texttt{DMK} \times \texttt{DTOS} \times \texttt{DIR} \times \texttt{NH} \times \texttt{NHB} \tag{34}$$

$$\times \texttt{SVLN} \times \texttt{DVLN} \times \texttt{ISMC} \times \texttt{ODMC} \times \texttt{IDMC} \times \texttt{OSMC} \times \texttt{MPLS1} \times ... \times \texttt{MPLS10} \tag{35}$$

$$\times \texttt{CL} \times \texttt{SL} \times \texttt{AL} \times \texttt{RA} \times \texttt{ENG} \times \texttt{EXID}. \tag{36}$$

Zbiory $\boldsymbol{F}_i$ przedstawione powyżej, zdefiniowane są dokumentacją `nfdump`. Przykład prostego kodowania wybranych parametrów pakietów przedstawia tabela 1. W rozważanym przypadku wymagane jest obserwowanie cech pakietów umożliwiających określenie ich adresów źródłowych i docelowych, znaczników czasowych oznaczających czas obserwacji pakietu oraz ilość przenoszonych danych.

$\boldsymbol{F}_i$	Parametr
$\mathbb{R}$	czas obserwacji pakietu
$\mathbb{Z}$	adres źródłowy
$\mathbb{Z}$	adres docelow
$\mathbb{Z}$	port źródłowy
$\mathbb{Z}$	port docelowy
$\mathbb{Z}$	protokół
$\mathbb{Z}$	rozmiar pakietu

Tabela 1: Reprezentacja parametrów pakietów.

Zbiór obserwowanych wartości pól nagłówków i parametrów pakietów jest nazywany ruchem sieciowym.

ruch sieciowy: Podzbiór dziedziny parametrów pakietów, $\mathbf{F}_0 \subset \mathbf{F}$, zrekonstruowany na podstawie zbioru zaobserwowanych pakietów sieciowych stanowi ruch sieciowy.

Z zaproponowanych powyżej definicji wynika oczywiście, że zaobserwowany pakiet sieciowy p jest elementem ruchu sieciowego,

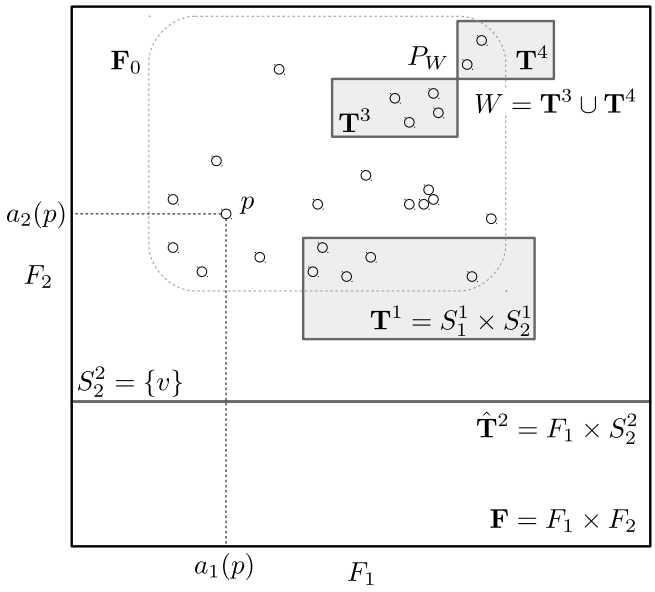
$$p \in \mathbf{F}_0 \subset \mathbf{F}. \tag{37}$$

Podstawowym obiektem analizowanym w zaprojektowanym systemie jest *przepływ pakietów sieciowych*. Definicja tego pojęcia wymaga odwołania się do koncepcji atrybutów, selektorów oraz termów, które zostaną wykorzystane w jej sformułowaniu.

atrybut pakietu: Funkcja postaci

$$a_i : \mathbf{F} \to F_i,\ i = 1, ..., n, \tag{38}$$

przypisująca liczbę ze zbioru $\boldsymbol{F}_i$ do pakietu $p \in \mathbf{F}$ nazywana będzie atrybutem pakietu.



Rysunek 11: Ilustracja dziedziny wnioskowania.

Jak łatwo zauważyć, atrybut jest formalnym narzędziem pozwalającym odczytać wartość wybranego parametru pakietu, a także reprezentować pakiet jako wektor:

$$p = (a_1(p), ..., a_n(p)) \in \mathbf{F}. \tag{39}$$

Na podstawie atrybutów zaobserwowanych pakietów sieciowych można zrekonstruować obrazu ruchu sieciowego:

$$F_i^0 = \bigcup_{p \in \mathbf{F}_0} \{a_i(p)\} \subset F_i, \ i = 1, ..., n. \tag{40}$$

W rezultacie ruch sieciowy jest zbiorem postaci:

$$\mathbf{F}_0 = F_1^0 \times ... \times F_n^0. \tag{41}$$

selektor: Podzbiór

$$S_i \subseteq F_i, \ i = 1, ..., n, \tag{42}$$

odpowiadający atrybutowi a_i nazywany będzie *selektorem*.

Selektory będą wykorzystane jako narzędzia umożliwiające określenie podzbiorów pakietów, których parametry spełniają zadane warunki:

$$a_i(p) \in S_i, \ i = 1, ..., n. \tag{43}$$

Domyślnie dla dowolnego atrybutu a_i przyjmowana będzie postać selektora $S_i \equiv F_i$.

term: Iloczyn kartezjański selektorów

$$\mathbf{T} = S_1 \times ... \times S_n \tag{44}$$

określa `term`.

Zgodnie z przyjętymi definicjami term $\mathbf{T}$ jest iloczynem logicznym warunków charakteryzujących pakiet sieciowy. Zachodzi wówczas następujący warunek:

$$p \in \mathbf{T} = S_1 \times ... \times S_n \equiv \bigwedge_{s=1}^{n} a_i(p) \in S_i. \tag{45}$$

Rozważmy term określony selektorami

$$S_i \times S_j = \{1025, ..., 2^{16} - 1\} \times \{80, 443\} \subset \mathtt{SP} \times \mathtt{DP},$$

przy czym $S_k = F_k$ dla $k \in \{1, ..., n\} \setminus \{i, j\}$. Term taki izoluje pakiety spełniające następujące warunki wyrażone w języku `pcap-filter`:

```
(src port > 1024) and (dst port 443 or dst port 80).
```

term bazowy: Niech $\hat{S}_i = \{v\}$ będzie selektorem pojedynczym określonym przez zbiór jednoelementowy. Iloczyn kartezjański

$$\hat{\mathbf{T}}(i) = F_1 \times ... \times \hat{S}_i \times ... \times F_n, \tag{46}$$

określa *term bazowy.*

Pojęcia termu oraz atrybutu mogą być wykorzystane do sformułowania definicji filtru pakietów. Rozważmy kolekcję termów:

$$\mathbf{T}^j = S_1^j \times ... \times S_n^j, \; j = 1, ..., m. \tag{47}$$

Selektory wykorzystane powyżej opisują pakiety charakteryzujące się określonymi wartościami pól nagłówków protokołów. Pakiet $p \in \mathbf{F}$ spełnia zadane selektorami warunki, jeżeli:

$$p \in \bigcup_{j=1}^{m} \mathbf{T}^j = \bigcup_{j=1}^{m} (S_1^j \times ... \times S_n^j). \tag{48}$$

Filtr pakietów spełniających warunki zadane przez kolekcję termów może być formalnie zdefiniowany przy pomocy funkcji wskaźnikowej.

filtr pakietów: Rozważmy term $W = \bigcup_{j=1}^{m} \mathbf{T}^j$ zdefiniowany przez sumę iloczynów kartezjańskich selektorów. Funkcja wskaźnikowa

$$\delta_W(p) = \begin{cases} 1, & p \in W, \\ 0, & p \notin W, \end{cases} \tag{49}$$

określa *filtr pakietów.*

Filtr pakietów δ_W, zdefiniowany dla termu W, pozwala wyizolować z ruchu sieciowego podzbiór pakietów tworzących przepływ.

przepływ pakietów: Zbiór zaobserwowanych pakietów spełniających warunki określone przez term W,

$$P[W] = \{p \in \mathbf{F}_0 | \delta_W(p) = 1\} \subset \mathbf{F}, \tag{50}$$

określa *przepływ pakietów.*

Przepływ pakietów, zgodnie z powyższą definicją, jest zbiorem pakietów. Zbiór taki może zostać scharakteryzowany liczbowo przez odpowiednio zaprojektowaną statystykę, która w rozważanym kontekście będzie charakteryzować właściwości przepływu pakietów.

liczbowa charakterystyka przepływu: Funkcja postaci:

$$g : \mathscr{P}(\mathbf{F}_0) \times \Theta \to \mathbb{R}, \tag{51}$$

określona dla dowolnego podzbioru pakietów ruchu sieciowego $W \subset \mathscr{P}(\mathbf{F}_0)$ oraz parametru $\theta \in \Theta$, wyznacza liczbową charakterystykę przepływu.

Podstawowe charakterystyki liczbowe zbiorów pakietów obserwowanych przez zadany filtr określają szybkość przepływu pakietów wyrażaną bitach lub pakietach na sekundę. Niech $h \in \mathbb{R},\ h > 0$, oznacza przedział czasu w którym zbierane są próbki pakietów. Rozważmy selektor $S_k(t) = (t-h, t] \subset F_k$ wybierający pakiety zaobserwowane w okresie próbkowania $(t-h, t]$ poprzedzającym chwilę $t \in \mathbb{R}$ oraz term

$$W(t) = \bigcup_{j=1}^{m} (S_1^j \times \ldots \times S_k(t) \times \ldots \times S_n^j). \tag{52}$$

Niech a_s oznacza rozmiar pakietu. W chwili t szybkość przepływu danych (bitrate) przenoszonych przez pakiety należące do zbioru $P[W(t)]$ jest wyznaczana przez funkcję:

$$g_{\mathtt{bps}}(P[W(t)], h) = \frac{1}{h} \sum_{p \in P[W(t)]} a_s(p). \tag{53}$$

Szybkość przepływu pakietów jest wyznaczana przez funkcję:

$$g_{\mathtt{pps}}(P[W(t)], h) = \frac{1}{h} |P[W(t)]|, \tag{54}$$

gdzie $|\bullet|$ oznacza moc zbioru.

reguła sterowania: Niech $\mathbf{U}$ oznacza zbiór wartości parametrów konfiguracji ograniczników szybkości transmisji pakietów (policerów, shaperów) urządzenia sieciowego (typu firewall).

Odpowiadajaca termowi W reguła sterowania

$$\mu_W : \mathbb{R}^m \to \mathbf{U} \tag{55}$$

przyporządkowuje wektorowi liczbowych charakterystyk przepływu parametry konfiguracji urządzenia sieciowego.

Reguła sterowania określa algorytm kształtowania dynamiki przepływu. Algortym ten wyznacza na podstawie wybranych atrybutów przepływu sygnał określający wartości parametrów pracy ograniczników szybkości transmisji pakietów (policerów). Poniżej przedstawiono przykład konfiguracji parametrów ogranicznika szybkości transmisji systemu Junos. Nastawy tłumika (policera), parametry `bandwidth-limit` oraz `burst-size-limit`, określają ograniczenia dla przepływu, do którego obsługi tłumik został przypisany:

```
set firewall policer fldx-qfx10k-dirty-AP
  if-exceeding bandwidth-limit 14m
set firewall policer fldx-qfx10k-dirty-AP
  if-exceeding burst-size-limit 2147450880
```

reguła filtracji: Niech $\mathbf{g}(P[W]) = [g_1(P[W]), ..., g_m(P[W])]^T \in \mathbb{R}^m$ oznacza wektor liczbowych charakterystyk przepływu $P[W]$. Reguła filtracji przypisuje regułę sterowania pakietom należącym do określonego przepływu:

$$\bigwedge_{p \in \mathbf{F}} p \in P[W] \rightarrow \mu_W(\mathbf{g}(P[W])). \tag{56}$$

Poniżej przedstawiono przykład konfiguracji reguły filtracji systemu Junos. Reguła filtracji została założona dla przepływu zdefiniowanego przez selektor określający adres źródłowy `a.b.c.d/32` przepływu o nazwie `fldx-qfx10k-dirty`. Pakietom należącym do tego przepływu przypisywana jest reguła sterowania obsługująca policer o nazwie `fldx-qfx10k-dirty-AP`:

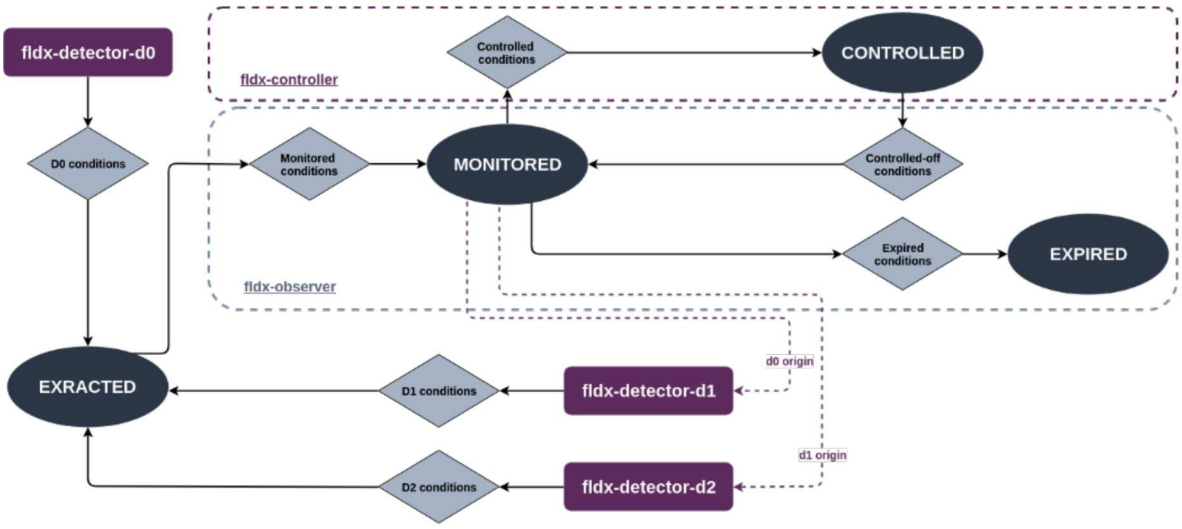
```
set firewall family inet filter fldx-qfx10k-dirty term
  fldx-qfx10k-dirty-AP-1
    from source-address a.b.c.d/32
set firewall family inet filter fldx-qfx10k-dirty term
  fldx-qfx10k-dirty-AP-1
    then policer fldx-qfx10k-dirty-AP
```

Omówione dalej konstrukcje mechanizmów detekcji anomalii i kształtowania przepływów wykorzystują zbudowany przeze mnie model dziedziny wnioskowania. Model ten został zaimplementowany w jądrze systemu `FLDX`.

Detekcja anomalii i synteza przepływów chronionych

Podstawowym obiektem w systemie `FLDX` jest przepływ określony podaną wyżej definicją. Wędrówka przepływu przez system opisana jest przez zaprojektowany przeze mnie automat skończony, którego graf przedstawiony został na Rysunku 12.

Algorytm detekcji anomalii identyfikuje w ruchu sieciowym zbiór przepływów bazowych oraz sortuje ten zbiór zgodnie z relacją porządkującą określającą znaczenie przepływu. Relacja



Rysunek 12: Graf stanów przepływu.

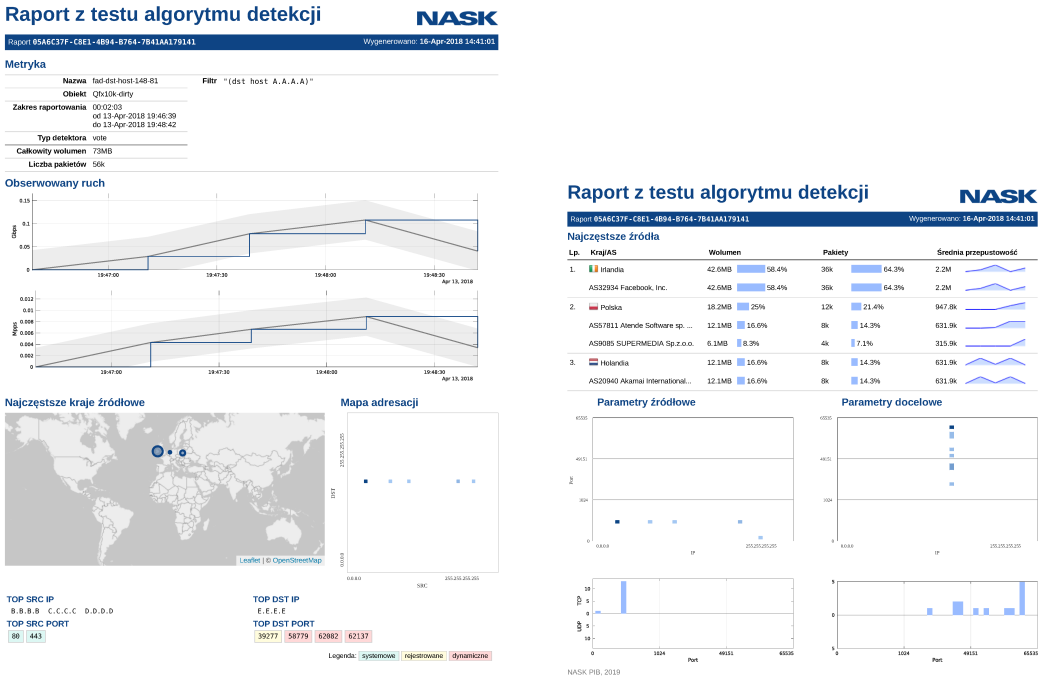
porządkująca zbiór przepływów bazowych jest konstruowana na podstawie agregacji kolekcji porządków cząstkowych określonych na atrybutach ruchu sieciowego (zaobserwowanym w okresie próbkowania). W podstawowej wersji systemu uwzględnianie są następujące:

- atrybuty: `dstip, dstport, srcip, srcport, proto`;
- kryteria uporządkowania: `bps, pps, flows, bytes, packets, duration`.

Ocena charakterystyki wyizolowanych przepływów jest przeprowadzana w kolejnych krokach wykorzystujących sygnał pochodne konstruowane na podstawie bezpośrednich obserwacji pakietów. Algorytm detekcji anomalii identyfikuje w zbiorze pakietów zaobserwowanych w okresie t selektory pakietów tworzących wyróżniające się wzorce komunikacji. Wartości atrybutów określanych przez zidentyfikowane selektory są wykorzystywane do określenia termów bazowych. Termy bazowe wykorzystywane są do definiowania przepływów bazowych, czyli zbiorów (obserwowanych) pakietów wymagających monitorowania i kształtowania. Na podstawie termów bazowych konstruowane są filtry definiujące przepływy chronione.

Synteza filtrów definiujących przepływy realizowana jest przez hierarchię detektorów systemu `FLDX`. Hierarchia ta zbudowana jest przez detektory kolejnych rzędów $k = 0,1,2$, przy czym detektory wyższych rzędów generują filtry o rosnącej rozdzielczości (precyzji izolacji przepływów). Detektory przeprowadzają syntezę rozwiązując odpowiednio sformułowane zadania analizy sygnałowej. Zadania te polegają na wyszukiwaniu w obserwowanym ruchu sieciowym $\mathbf{F}_0$ szeregów czasowych X_k, $k = 0,1,2$, złożonych z liczbowych charakterystyk przepływów, spełniających następujący układ równań:

$$\begin{aligned} X_0 &= \mathscr{C}(\mathbf{F}_0), \\ X_1 &= \mathscr{F}(\mathbf{F}_0, W_0), \quad \text{gdzie} \quad X_0 \approx W_0^{-1} X_1, \\ X_2 &= \mathscr{F}(\mathbf{F}_0, W_1), \quad \text{gdzie} \quad X_1 \approx W_1^{-1} X_2. \end{aligned} \tag{57}$$



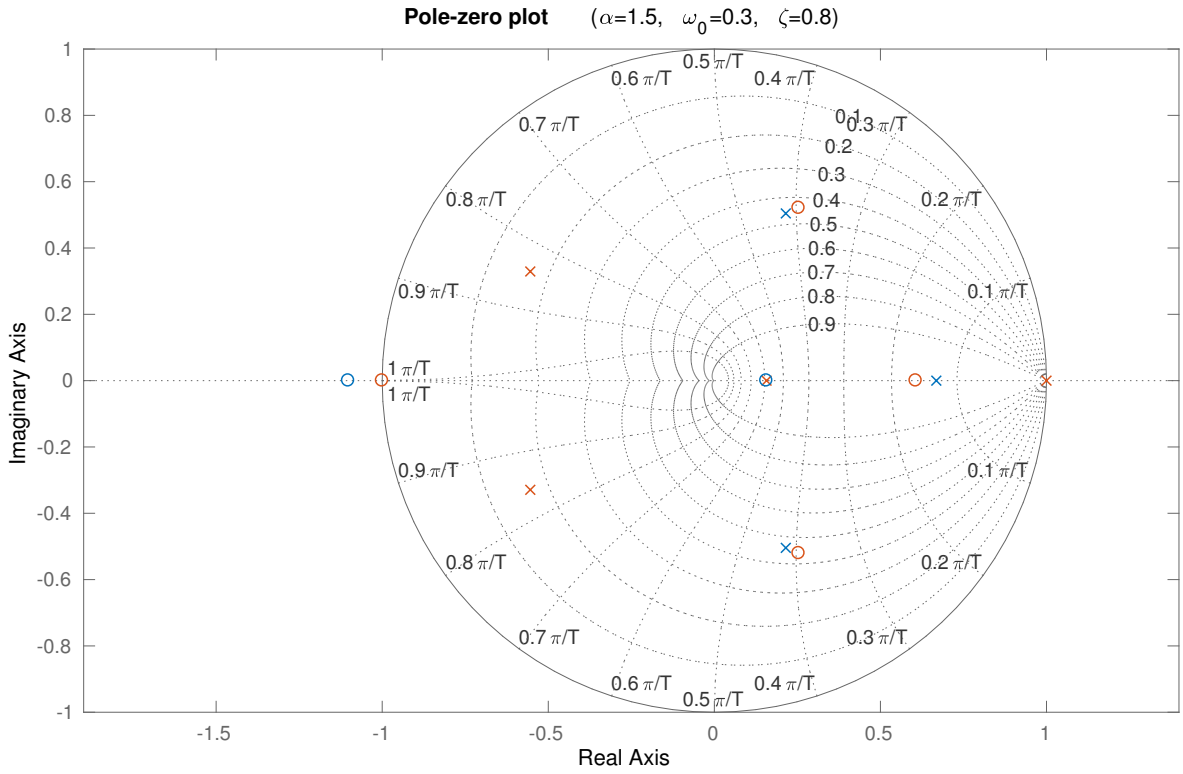
Rysunek 13: Analiza przepływu chronionego.

W powyższym ogólnym sformułowaniu zadania syntezy $\mathscr{C}$ oznacza operację wyboru przepływów bazowych. We wdrażanej wersji systemu operator $\mathscr{C}$ wykorzystuje domyślnie algorytm Bordy obliczający relację porządku agregującą kryteria oceny istotności przepływu. Relacja ta spełnia warunek Parety, nie-autorytarności oraz uniwersalności, lecz narusza warunek niezależności od wariantów nieistotnych. Operator $\mathscr{C}$ może wykorzystywać dowolne algorytmy agregacji porządków. Operator $\mathscr{F}$ wyszukuje w obserwowanym ruchu przepływy określone przez operator W_k mieszania szeregów czasowych charakteryzujących przepływy niższego rzędu. Takie sformułowanie zadania detekcji prowadzi do następującej rekonstrukcji sygnałów składowych ataku:

$$X_k = (\tilde{W}_{k-1} \circ \cdots \circ \tilde{W}_0)(X_0), \tag{58}$$

gdzie $\tilde{W}_k$ oznacza odpowiednio skonstruowany rzut operatora W_k mieszania sygnałów. Zgodnie z przedstawioną wyżej autorską koncepcją matematyczną, przepływy izolowane przez system `FLDX` konstruowane są na bazie kombinacji szeregów czasowych związanych z przepływami bazowymi. Przepływy wyższych rzędów, X_k, $k = 1,2$, wyznaczane są przez definicje filtrów pakietów o rosnącej precyzji filtracji.

Z każdym zarejestrowanym przepływem związana jest kolekcja liczników zdarzeń polegających na przekroczeniu przez sygnał zdefiniowanych norm. Suma wartości wytworzonych w ten sposób sygnałów binarnych może stanowić dla operatora sygnał alarmowy.



Rysunek 14: Rozkład zer i biegunów ukształtowanego modelu przepływu.

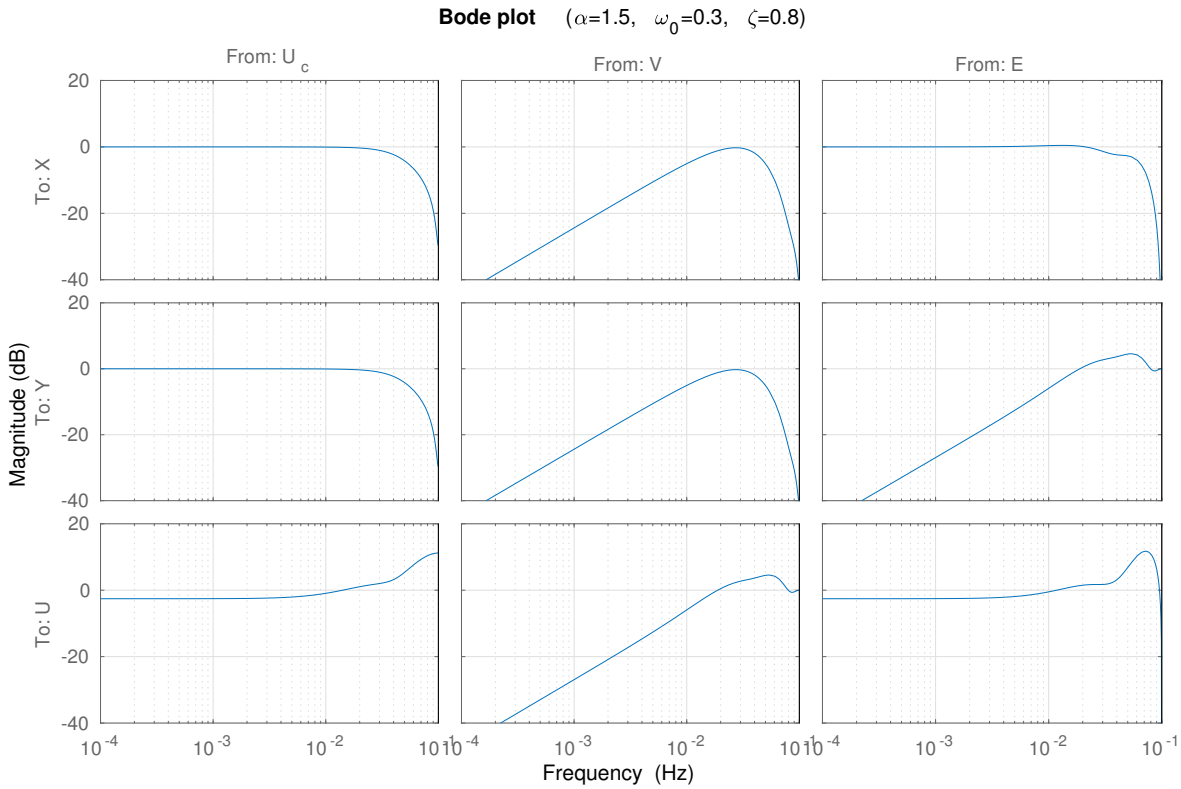
Kształtowanie dynamiki przepływów

Dynamika każdego przepływu chronionego kształtowana jest przez adaptacyjny regulator (R,T,S), którego synteza prowadzona jest w czasie rzeczywistym przez system `FLDX` na podstawie bieżących obserwacji ruchu sieciowego. Synteza algorytmów sterowania jest przeprowadzana przy pomocy autorskich mechanizmów opracowanych na podstawie wyników badań prowadzonych przez NASK PIB w obszarze teorii sterowania. Równania syntezy wyprowadzane są dla modelu wielomianowego:

$$\begin{aligned} A(q)x(t) &= B(q)(u(t)+v(t)),\\ y(t) &= x(t)+e(t),\\ R(q)u(t) &= T(q)u_c(t)-S(q)y(t), \end{aligned} \tag{59}$$

gdzie q oznacza operator przesunięcia. Reguły sterowania, otrzymane w wyniku syntezy algorytmów (R,T,S), powiązane są z każdym przepływem chronionym. Model procesu (A,B) jest identyfikowany metodą filtru Kalmana przez następujący algorytm:

$$\begin{aligned} \varphi(k-1) &= [-y(k-1),\ldots,-y(t-n),u(t-d),\ldots,u(t-d-m)],\\ K(k) &= \frac{P(k-1)\varphi(k-1)}{\sigma^2+\varphi^T(k-1)P(k-1)\varphi(k-1)},\\ P(k) &= (I-K(k)\varphi^T(k-1))P(k-1)+V. \end{aligned} \tag{60}$$

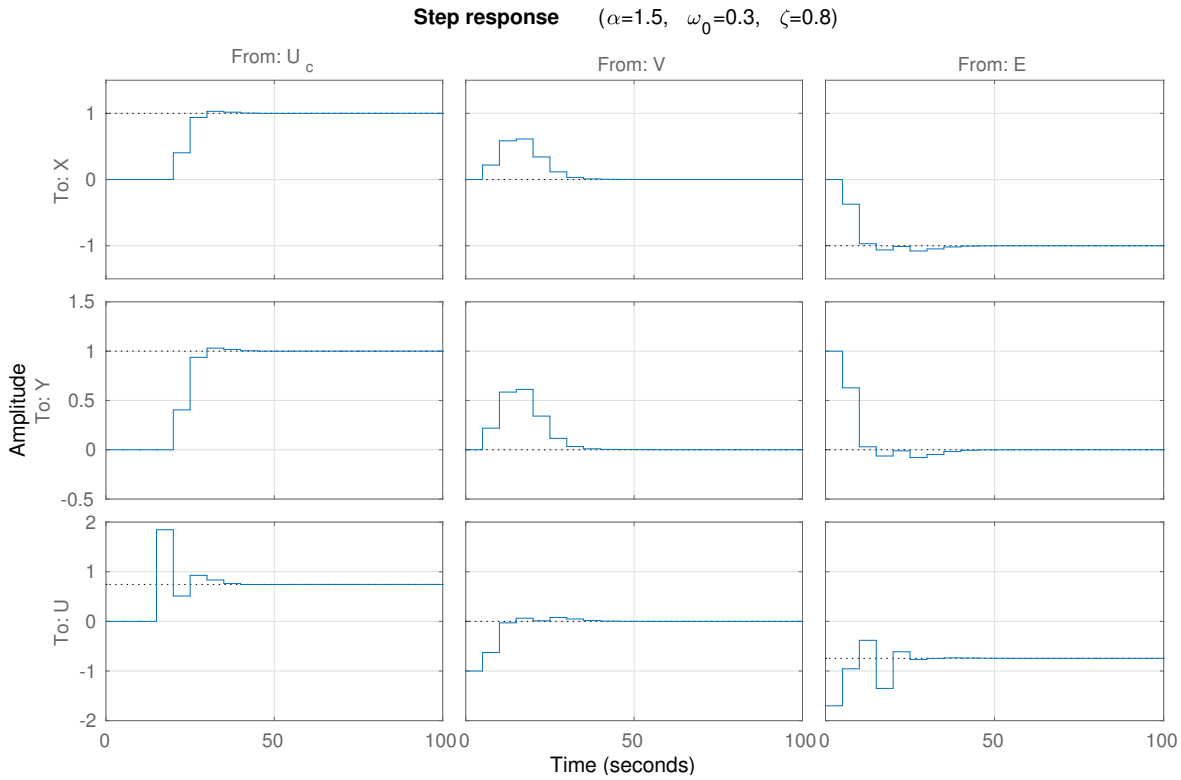


Rysunek 15: Charakterystyka amplitudowa Bodego zaprojektowanego układu sterowania.

Współczynniki modelu dostrajane są do obserwacji, tzn.:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)(y(k) - \varphi^T(k-1)\hat{\theta}(k-1)). \tag{61}$$

Wyniki testów laboratoryjnych działającego systemu sterownia adaptacyjnego przedstawiają Rysunki 14-17. Ilustracje przedstawiają charakterystyki zaprojektowanych automatycznie filtrów (R,T,S), dla identyfikowanego dynamicznie modelu (A,B), a także przebieg procesu tłumienia wygenerowanego sztucznie ataku oraz kształtowania dynamiki przepływu chronionego, zanieczyszczonego strumieniem pakietów będącym składową ataku.



Rysunek 16: Odpowiedź skokowa zaprojektowanego układu sterowania.

4.2 Adaptacyjny sterownik procesora dla jądra systemu Linux

Kolejnym niezwykle ważnym dla mnie osiągnięciem inżynierskim jest projekt i implementacja adaptacyjnego energooszczędnego sterownika procesora w środowisku jądra systemu Linux.

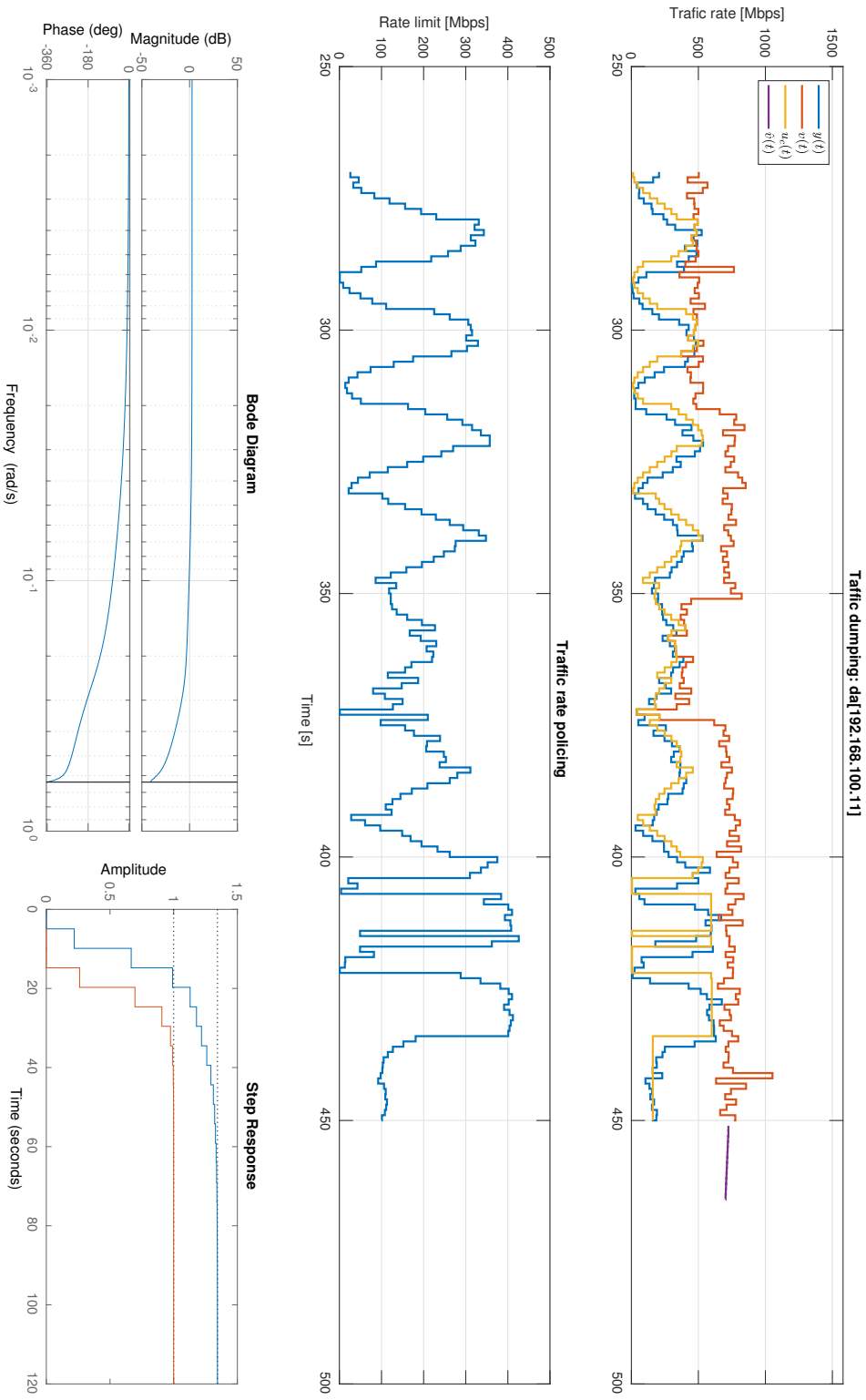
Zakładam, że dynamika procesu obliczeniowego na poziomie CPU może być opisana następującym układem równań różnicowych:

$$\begin{aligned} A(q)x(t) &= B(q)(u(t)+v(t)), \\ y(t) &= x(t)+e(t), \end{aligned} \tag{62}$$

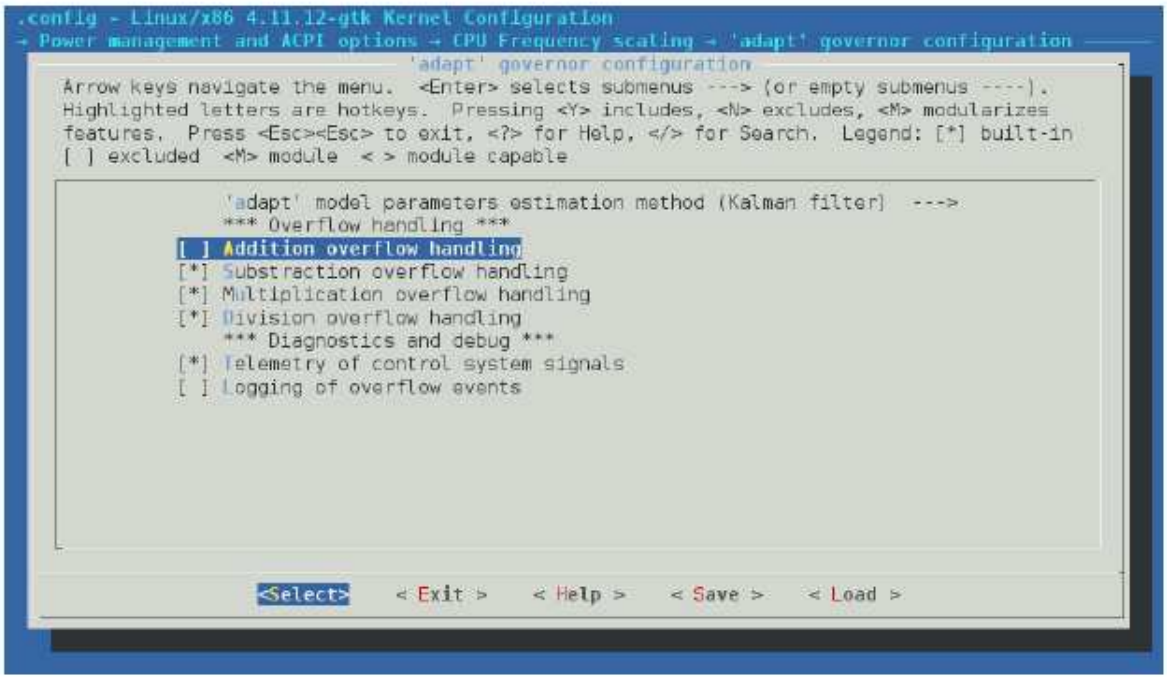
gdzie $x(t)$ jest kształtowanym sygnałem wyjściowym, $y(t)$ oznacza obserwowany sygnał pomiaru wartości wyjścia, $e(t)$ oznacza sygnał zakłócający pomiar sygnału $x(t)$, $u(t)$ oznacza sygnał sterowania, natomiast $v(t)$ nieznane wejście swobodne.

Dla modelu tego można w układzie adaptacyjnym obliczyć algorytm sterowania (R,T,S) kształtujący dynamikę obserwowanego procesu. Realizacja tak obliczonych filtrów wymaga jednak uwzględnienia ograniczeń związanych z obliczeniami stałopozycyjnymi oraz dokładnością numeryczną dostępną na poziomie jądra systemu operacyjnego. Ograniczenia te można opisać równaniem:

$$u(k) = g(w(k)), \tag{63}$$



Rysunek 17: Zapis przebiegu eksperymentu w laboratorium.



Rysunek 18: Zaimplementowany adaptacyjny energooszczędny sterownik procesora w menu konfiguracji parametrów kompilacji jądra systemu Linux.

gdzie $w(k)$ jest sygnałem obliczonym przez algorytm sterowania, a $u(k)$ sygnałem przykładanym do obiektu sterowania. Funkcja $g\colon \mathbb{R} \to \mathbb{P}$ rzutuje sygnał obliczony na przestrzeń sterowań dopuszczalnych, dla $\mathbb{P} = \{p_0, p_1, \ldots, p_n\} \subset \mathbb{R}$ oznaczającego zbiór stanów ACPI procesora. W jądrze systemu Linux dostępna jest następująca operacja:

$$p = g(x) = \min\left\{\arg\min_{p\in\mathbb{P}} |x - p|\right\}. \tag{64}$$

Ostatecznie algorytm sterowania działajacy na poziomie sterownika CPU w jądrze systemu Linux ma następującą wyprowadzoną przeze mnie postać (warto porównać ją ze znananą postacią regulatora całkującego z mechanizmem anti-windup [2]):

$$\begin{aligned} A_o(q)w(k) &= T(q)u_c(k) - S(q)y(k) + (A_o(q) - R(q))u(k) \\ u(k) &= g(w(k)). \end{aligned} \tag{65}$$

Implementacja algorytmu syntezy na poziomie jądra wymaga rozwiązania równania diofantycznego:

$$C\overline{R} + D\overline{S} = E, \tag{66}$$

gdzie:

$$\begin{aligned} C &= AR_d \\ D &= B^{-}S_d \\ E &= A_f A_o A_m. \end{aligned} \tag{67}$$

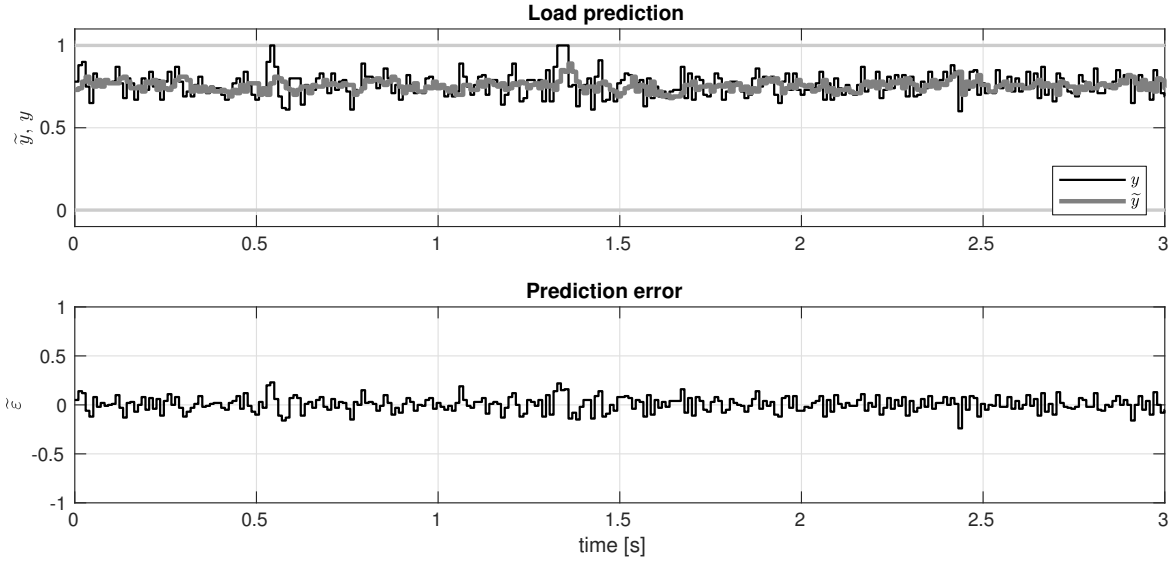
Zadanie to można sprowadzić do zadania rozwiązania układu równań:

$$\boldsymbol{M}\boldsymbol{x} = \boldsymbol{c}, \tag{68}$$

gdzie

$$\boldsymbol{M} = \underbrace{\begin{bmatrix} c_0 & & 0 \\ \vdots & \ddots & c_0 \\ \vdots & & \vdots \\ c_{n_c} & & \vdots \\ 0 & \ddots & c_{n_c} \end{bmatrix}}_{n_{\bar{r}}+1 \text{ columns}} \underbrace{\begin{bmatrix} d_0 & & 0 \\ \vdots & \ddots & d_0 \\ \vdots & & \vdots \\ d_{n_d} & & \vdots \\ 0 & \ddots & d_{n_d} \end{bmatrix}}_{n_{\bar{s}}+1 \text{ columns}}, \tag{69}$$

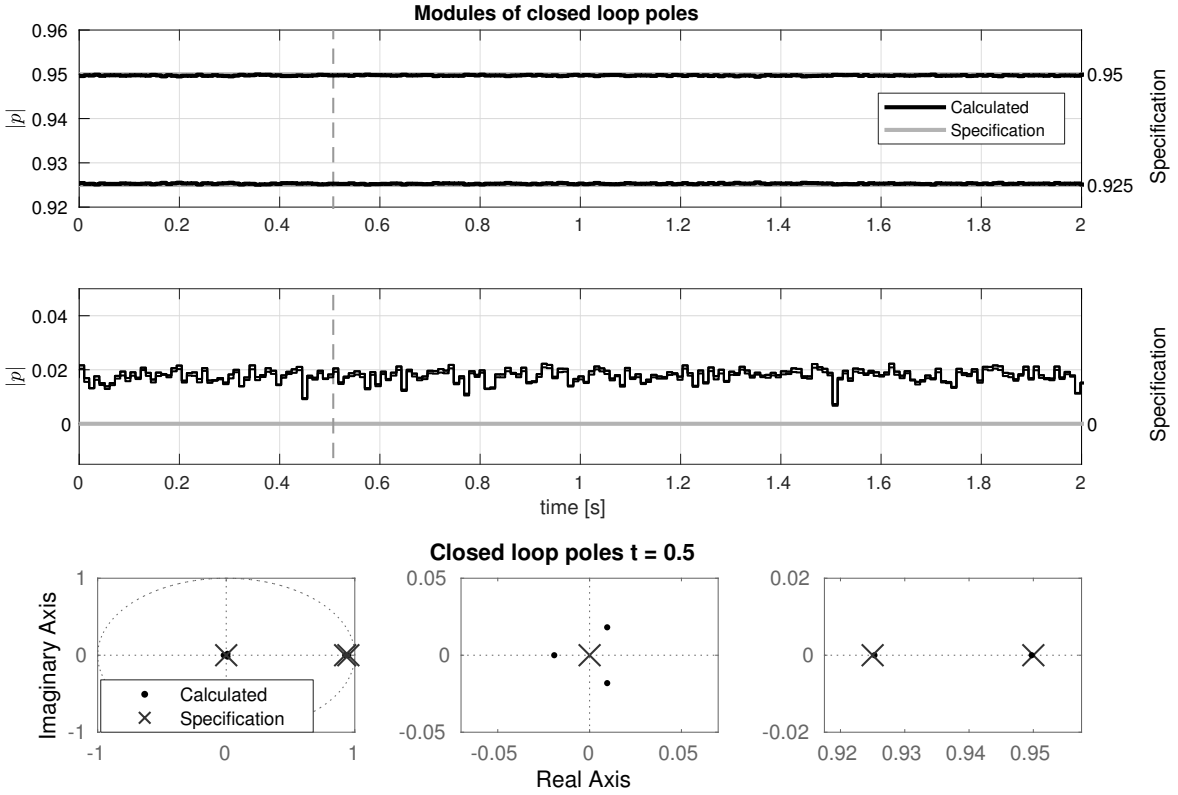
wektor $\boldsymbol{c}$ określa współczynniki wielomianu $\boldsymbol{C}$, a $\boldsymbol{x} = [\bar{r}_0, \bar{r}_1, \ldots, \bar{r}_{n_{\bar{r}}}, \bar{s}_0, \bar{s}_1, \ldots, \bar{s}_{n_{\bar{s}}}]$.



Rysunek 19: Prognoza obciążenia CPU metodą RLS realizowana na poziomie jądra systemu Linux.

Algorytm rozwiązywania powyższego układu równań zaimplementowany w środowisku jądra systemu Linux. Poniżej zaprezentowane zostały wyniki eksperymentów pokazujące pracę sterownika dla nastaw:

$$\begin{aligned} A_f(q) &= 1, \\ A_o(q) &= (q - 0.925)q^2, \\ A_m(q) &= (q - 0.95)q, \\ R_d(q) &= q + 1, \\ S_d(q) &= q - 1. \end{aligned} \tag{70}$$

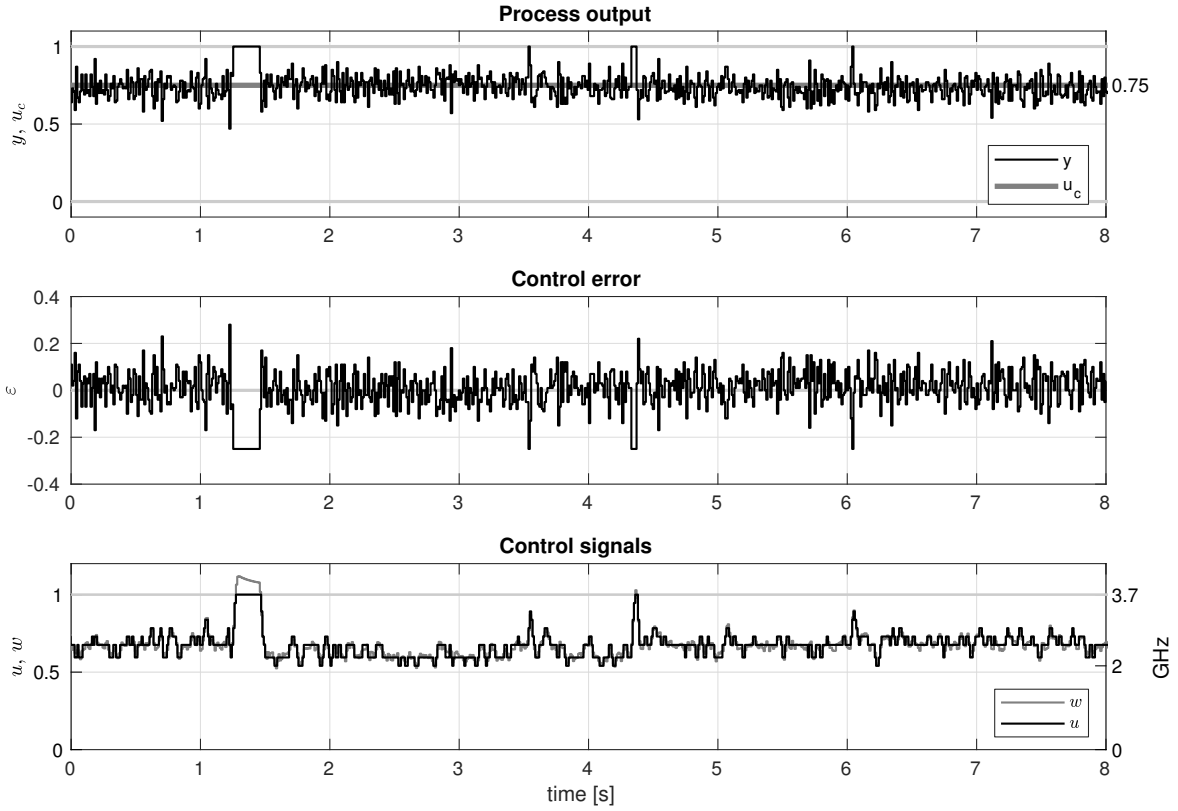


Rysunek 20: Dynamika procesu lokowania biegunów

4.3 `ARAKIS`: system wczesnego ostrzegania o incydentach cyberbezpieczeństwa

Moja działalność naukowo-badawcza związana jest również z rozwojem, wdrażaniem i utrzymaniem systemu wczesnego ostrzegania o incydentach bezpieczeństwa cybernetycznego `ARAKIS`. Kierując Zakładem Inżynierii Systemów Informatycznych w Pionie Badań i Rozwoju NASK PIB reazlizuję w tym zakresie m.in. nastepujące zadania:

- kierowanie procesem rozwoju i wdrażania systemu,
- kierowanie zespołem programistów i inżynierów wytwarzajacych system,
- projektowanie planu rozwoju architektury systemu,
- rozwój koncepcji mechanizmów analizy danych,
- koordynacja wdrożeń kolejnych wersji systemu,
- koordynacja prac badawczo-rozwojowych,
- wsparcie procesów budowy produktu i negocjacji biznesowych.



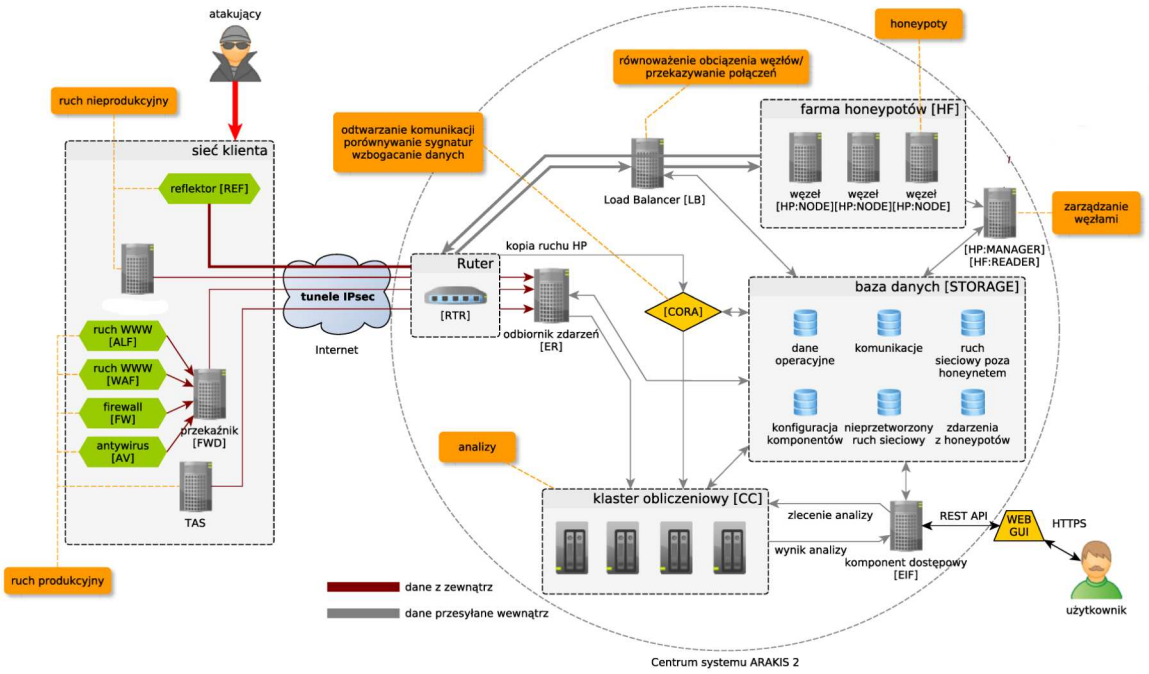
Rysunek 21: Ilustracja jakości regulacji (poniżej).

System ARAKIS stanowi element rządowego programu o tej samej nazwie, mającego zapewniać cyberbezpieczeństwo operatorów usług kluczowych oraz infrastruktury krytycznej państwa. Zakres i rola programu zostały wskazane w Ustawie o Krajowym Systemie Cyberbezpieczeństwa. Dokumentacja systemu jest w dużej części niejawna.

System `ARAKIS` rozwijany jest przez NASK PIB od 2012 roku, a pod moim kierownictwem od 2016 roku. Główna funkcjonalność systemu to przede wszystkim wykrywanie wzorców zaawansowanych ataków i zagrożeń występujących w sieci na podstawie agregacji i korelacji danych, generowanie opisu zaobserwowanych incydentów w postaci alarmów oraz sygnatur zarejestrowanych komunikacji. `ARAKIS` udostępnienia graficzny interfejs użytkownikom dzięki któremu możliwe jest łatwe zarządzanie i administrowanie systemem, przeglądanie oraz analizowanie zagrożeń. W systemie zaprojektowany został specjalny język zapytań AQL (ARAKIS Query Language) umożliwiający pozyskiwanie oraz prezentowanie danych w sposób dostosowany do indywidualnych potrzeb użytkownika. Architektura systemu w składa się z centrum, w którym wykonywane są złożone analizy i gromadzone dane, oraz sond zewnętrznych rozmieszczonych w różnych segmentach sieci klienta, odpowiedzialnych za dostarczenie danych do centrum.

Wyróżnia się następujące rodzaje sond będących źródłami danych w systemie:

- **reflektor**: sensor odpowiedzialny za utrzymywanie komunikacji pomiędzy odpowiednio

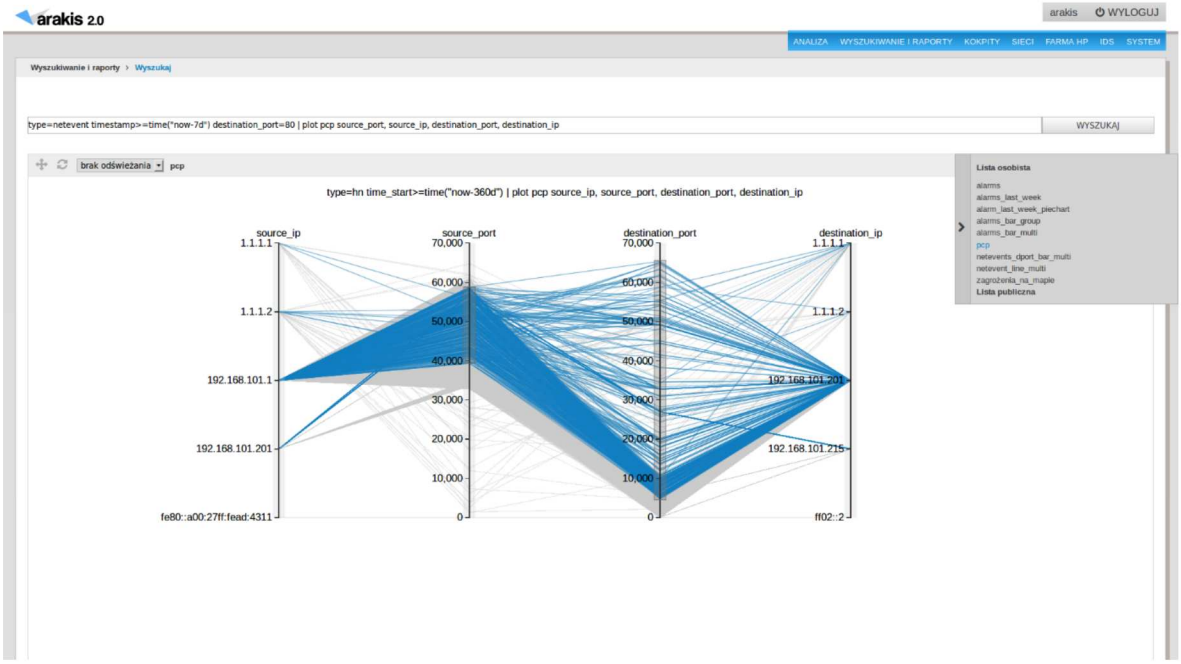


Rysunek 22: Schemat architektury systemu ARAKIS.

skonfigurowanymi adresami usług pułapek i farmą honeypotów zbierającą dane na temat rejestrowanych zagrożeń oraz nadużyć w sieci chronionej,

- **forwarder**: sensor odbierający i przekazujący logi zewnętrznych systemów bezpieczeństwa, m.in. zapór sieciowych (firewalli) oraz systemów antywirusowych,
- **TAS**: sensor odpowiedzialny za wykrywanie w monitorowanym ruchu niebezpiecznych komunikacji, m.in. z sieciami botów (botnet), oraz przesyłanie informacji o podejrzanych zdarzeniach do centrum systemu ARAKIS,
- **sensor SCADA**: sensor odpowiedzialny za utrzymywanie komunikacji z siecią pułapek wyposażonych w emulowane usługi systemów automatyki przemysłowej.

System obejmuje swoja ochrona znaczną część obszaru Polski, jest wykorzystywany do monitorowania pracy sieci instytucji infrastruktury krytycznej oraz administracji publicznej. Efektywne wykrywanie incydentów zagrażających cyberbezpieczeństwu chronionej sieci wymagało sprawnie współdziałających mechanizmów agregacji i korelacji danych rejestrowanych przez sieć sond. Stworzenie tego systemu to efekt pracy zespołu badawczego nad wieloma odrębnymi zagadnieniami technicznymi i naukowymi, między innymi: opracowanie algorytmów wykrywających w czasie wielomianowym podciągi znaków efektywnie charakteryzujące wzorce komunikacji skierowanych do sieci pułapek, zaprojektowanie mechanizmu uczenia maszynowego do tworzenia probabilistycznego modelu zapytań kierowanych do serwera oraz opracowanie procesu testowania hipotez oceniających zgodność tych zapytań z wykrytym wzorcem.



Rysunek 23: Przykład analizy połączeń.

Spis literatury

[1] P. Arabas and M. Karpowicz. Wykorzystanie informacji z rejestrów procesora do identyfikacji modelu poboru mocy przez serwer. *Przegląd Elektrotechniczny*, 92(3):34–41, 2016.

[2] K. J. Åström and B. Wittenmark. *Computer-controlled systems: theory and design*. Dover Publications, Mineola, NY, 2011.

[3] K. J. Åström and B. Wittenmark. *Adaptive control*. Dover Publications, Mineola, NY, 2013.

[4] J. Blackledge. Application of the fractional diffusion equation for predicting market behaviour. 2010.

[5] J. Gondzio, P. González-Brevis, and P. Munari. New developments in the primal–dual column generation technique . *European Journal of Operational Research*, 224(1):41–51, 2013.

[6] R. Johari, S. Mannor, and J. N. Tsitsiklis. Efficiency Loss in a Network Resource Allocation Game: The Case of Elastic Supply. *IEEE Transactions on Automatic Control*, 50(11):1712–1724, November 2005.

[7] M. P. Karpowicz and P. Arabas. Preliminary results on the Linux libpcap model identification. In *20th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 1056–1061. IEEE, 2015.

[8] F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8(1):33–37, 1997.

[9] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control for communication networks: Shadow prices, proportional fairness, and stability. *Journal of the Operational Research Society*, 49(3):237–252, 1998.

[10] A. Kozakiewicz and K. Malinowski. Network traffic routing using effective bandwidth theory. *European Transactions on Telecommunications*, 20(7):660–667, 2009.

[11] R. J. La and V. Anantharam. Charge-Sensitive TCP and Rate Control in the Internet. In *IEEE INFOCOM 2000, Tel-Aviv, Israel*, pages 1166–1175, 2000.

[12] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of Ethernet traffic (extended version). *Networking, IEEE/ACM Transactions on*, 2(1):1–15, Feb 1994.

[13] S. H. Low. A duality model of TCP and queue management algorithms. *IEEE/ACM Transactions on Networking*, 11(4):525–536, August 2003.

[14] S. H. Low and D. E. Lapsley. Optimization Flow Control, I: Basic Algorithm and Convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, December 1999.

[15] S. H. Low, F. Paganini, J. Wang, and J. C. Doyle. Linear stability of TCP/RED and a scalable control* 1. *Computer Networks*, 43(5):633–647, 2003.

[16] K. Malinowski. Optimization network flow control and price coordination with feedback: Proposal of a new distributed algorithm. *Computer Communications*, 25(11-12):1028–1036, July 2002.

[17] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, 2000.

[18] R. Srikant. *The Mathematics of Internet Congestion Control.* Birkhäuser Boston, December 2003.

[19] J. Stiglitz. *Making globalization work.* Penguin Books, London, 2006.

[20] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. *Networking, IEEE/ACM Transactions on*, 5(1):71–86, Feb 1997.