

Streszczenie

W tej rozprawie rozważamy problemy kolejkowania pakietów online w różnych modelach obliczeń rozproszonych w sieciach komputerowych. W szczególności adresujemy problemy związane z podatnością kanałów komunikacyjnych na błędy i z interferencjami wywołanymi przez inne transmisje.

Rozważane algorytmy wyróżniają się tym, że pakiety pojawiają się w systemie wraz z upływem czasu. Wstrzykiwania pakietów są kontrolowane przez adwersarza lub przez proces stochastyczny. Projektujemy efektywne algorytmiczne rozwiązania dla rozważanych modeli oraz pokazujemy wyniki niemożliwości. Większość naszych algorytmów jest stabilna, czyli są w stanie dostarczać pakiety w takim tempie, w jakim się one pojawiają w systemie; w rezultacie kolejki pakietów są zawsze ograniczone.

W rozdziale 2 badamy zawodne połączenie składające się z wielu kanałów komunikacyjnych. Rozszerzamy model z pracy [F. Anta i in., Measuring the impact of adversarial errors on packet scheduling strategies, J. Scheduling 2016] poprzez wprowadzenie błędów na niezależnych kanałach i błędów na równoległych kanałach. Dla obu typów kanałów podajemy algorytmy oraz dowody ich optymalności. Dla niezależnych kanałów optymalna konkurencyjna przepustowość wynosi $1/2$. Dla równoległych kanałów optymalna konkurencyjna przepustowość zbiega do $3/4$ wraz z rosnącą liczbą kanałów w sytuacji, gdy rozważane błędy to równoległe zakłócenia. Dla poważniejszych błędów typu crash podajemy algorytm i pasujący wynik niemożliwości -- konkurencyjna przepustowość zbiega do wartości złotego podziału $\frac{1+\sqrt{5}}{2}$ wraz z rosnącą liczbą kanałów. Dość zaskakujące jest to, że w modelu z równoległymi błędami, w przeciwieństwie do modelu z niezależnymi kanałami, optymalna konkurencyjna przepustowość z zakłóceniami jest inna niż z błędami typu crash. Nasze metody dla modelu z zakłóceniami są zupełnie inne niż metody dla modelu z błędami typu crash. Na koniec budujemy algorytm, który osiąga konkurencyjną przepustowość 1 na pojedynczym kanale, o ile działa z przyspieszeniem równym co najmniej 2.

W rozdziale 3 zajmujemy się dobrze znanym modelem MAC. Budujemy algorytm, który osiąga stabilność przeciwko adwersarzowi typu leaky bucket, nawet przy bardzo ograniczonej informacji z kanału. Skupiamy się na słabych urządzeniach -- z ograniczoną pamięcią i ograniczoną możliwością otrzymywania informacji z kanału komunikacyjnego. Pokazujemy, że istnieje lokalny i adaptacyjny algorytm kolejkowania, który korzysta z niewielkiej ilości pamięci, a mimo to jest uniwersalnie stabilny -- gwarantuje ograniczone kolejki dla każdego tempa wstrzykiwania $\rho < 1$. Pokazaliśmy też, że kiedy algorytmy w ogóle nie mają dostępu do pamięci pomocniczej, maksymalne tempo wstrzykiwań, dla których da się zagwarantować stabilność, wynosi $O(1/\log n)$. Dla nieco mniejszego tempa wstrzykiwań $c/\log^2 n$ (dla stałej $c > 0$) budujemy lokalny nieadaptacyjny algorytm kolejkowania pakietów bez użycia pamięci opracowany na podstawie kombinatorycznej koncepcji uniwersalnie silnych selektorów.

W rozdziale 4 przechodzimy do uogólnienia modelu MAC -- kolizje między nadawcami będą modelowane przez dowolne grafy (zamiast tylko przez kliki). Podajemy deterministyczny algorytm, który osiąga uniwersalną stabilność i jest adwersarialnym odpowiednikiem wyniku z pracy [Shah i in., Medium Access Using Queues, FOCS11], gdzie podano algorytm osiągający stochastyczną stabilność. Pokazujemy, że nasz algorytm jest optymalny, czyli żaden algorytm nie

jest w stanie zagwarantować stabilności dla tempa wstrzykiwań $\rho=1$. Przewagą naszego rozwiązania nad rozwiązaniem Shah i in. jest to, że nasz algorytm działa dla bardzo ograniczonej informacji z kanału -- bazuje tylko na potwierdzeniach udanego nadania wiadomości. Dla porównania, wynik Shah i in. wymaga szczegółowych informacji z kanału dotyczących transmisji sąsiadów danego urządzenia. Co więcej, wariant naszego algorytmu jest stochastycznie stabilny, choć wymaga informacji o tempie wstrzykiwań $\rho < 1$.

Szczególną uwagę w naszej pracy zwracamy na słabe urządzenia z niewielką pamięcią i mocą obliczeniową oraz z ograniczonymi informacjami z kanału komunikacji. Podczas opracowywania naszych wyników zbudowaliśmy kilka ogólnych narzędzi, które mogą znaleźć zastosowania w innym kontekście. W szczególności wprowadziliśmy uniwersalnie silne selektory i pojemnościowe selektory. Poza tym rozszerzyliśmy techniki analizy online algorytmów rozproszonych.