

## **KOD FONTANNOWY Z PRZYRASTAJĄCĄ LICZBĄ SYMBOLI ŹRÓDŁOWYCH** FOUNTAIN CODE WITH INCREASING NUMBER OF SOURCE SYMBOLS

Kamila Haustein<sup>1</sup>; Marek Blok<sup>2</sup>

<sup>1</sup> Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, Gdańsk, hausteinkamila@gmail.com

<sup>2</sup> Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, Gdańsk, marek.blok@pg.edu.pl

**Streszczenie:** Kody fontannowe zabezpieczające transmisję przed wymazaniami wyróżnia brak określonej z góry długości i losowy sposób generacji kolejnych pakietów kodowych. W pracy pokazano jak właściwości tych kodów zależą od rozkładu macierzy generującej dla dwóch wariantów dostarczania pakietów do kodera. Szczególnie interesujący jest przypadek, gdy pakiety źródłowe trafiają do kodera stopniowo podczas transmisji. Uzyskane wyniki wskazują na możliwość realizacji transmisji strumieniowej o ograniczonej złożoności i wymaganiach pamięciowych dekodera.

**Abstract:** Fountain codes that protect the transmission against erasures are distinguished by the lack of a predetermined length and the random method of generation of successive code packets. The paper shows how the properties of these codes depend on the distribution of the generating matrix for two variants of packet delivery to the encoder. Of particular interest is the case where the source packets reach the encoder gradually during transmission. The obtained results indicate the possibility of stream transmission with limited complexity and memory requirements of the decoder.

**Słowa kluczowe:** nadmiarowe kodowanie korekcyjne, kanał z wymazywaniem, kody fontannowe, transmisja strumieniowa.

**Keywords:** erasure channel, error correction coding, fountain codes, stream transmission.

### 1. WSTĘP

W dzisiejszych czasach do przesyłania wiadomości korzystamy z technologii takich jak Internet czy telefon komórkowy. Podczas transmisji przez kanał mogą występować błędy, które wynikają ze zniekształceń i zakłóceń. Powoduje to, że część pakietów koniecznych do przesłania zostaje wymazana i utracona, co powoduje, że nie dotrą one do dekodera. Poprzez zastosowanie kodów fontannowych mamy możliwość odtworzenia wymazanych pakietów i poprawne przesłanie wiadomości bez względu na warunki panujące w kanale. Kody te nie mają ograniczonej długości i nadawane pakiety kodowe generowane są aż do momentu, w którym dekodery jest w stanie odzyskać przekazywane dane źródłowe.

Koncepcja kodów fontannowych zakłada, że informacja o pakietach źródłowych jest rozpraszana w wielu pakietach kodowych, tak żeby po stronie odbiorczej możliwe było prawidłowe odtworzenie wiadomości źródłowej na podstawie jak najmniejszego podzbioru nadawanych pakietów pomimo losowej utraty niektó-

rych pakietów podczas transmisji. Istnieje kilka rodzajów kodów fontannowych takich jak kody Tornado [4], kody LT [4] oraz kody Raptor [4] i ich usprawniona wersja kody RaptorQ [7].

Podstawową cechą kodów fontannowych jest, to że bieżący symbol kodowy jest generowany jako suma modulo dwa losowo wybranych symboli źródłowych. Z kolei po stronie odbiorczej odtworzenie wybranego symbolu źródłowego wymaga odnalezienia takiego zestawu symboli kodowych, których suma modulo dwa jest równa odtwarzanemu symbolowi źródłowemu. Taka konstrukcja oznacza, że po stronie nadawczej zawsze możemy wygenerować kolejny symbol kodowy, a po stronie nadawczej ma znaczenie przede wszystkim, ile symboli kodowych udało się odebrać, a niektóre z nich zostały wymazane w kanale. Koncepcję tę łatwo przenieść z kodowania symboli dla kanału z wymazywaniem do kodowania pakietów dla kanału z wymazywaniem pakietów.

Chociaż kody fontannowe gwarantują, że przesyłana wiadomość zostanie poprawnie odebrana niezależnie od prawdopodobieństwa wymazywania, to dobór sposobu losowego generowania kolumny macierzy generującej kodu wskazującej pakiety źródłowe tworzące pakiet kodowy wpływa na liczbę pakietów kodowych potrzebnych do zdekodowania przekazywanej wiadomości oraz opóźnienia dekodowania poszczególnych jej pakietów. Przeprowadzone badania miały na celu zweryfikowanie wpływu macierzy generującej na pracę dekodera dla dwóch wariantów: macierzy o równomiernym rozkładzie jedynek oraz macierzy o wagach kolumn losowanych zgodnie z rozkładem solitonowym [8].

Kolejny analizowany w pracy aspekt, dotyczy sposobu dostarczania danych do kodera w nadajniku. Podstawowa koncepcja kodów fontannowych zakłada, że koder w nadajniku zna od razu wszystkie przekazywane symbole/pakiety źródłowe. W pracy rozpatrujemy dodatkowo wariant, w którym kolejne pakiety źródłowe trafiają do kodera stopniowo i koder generuje pakiety kodowe na bieżąco jeszcze za nim otrzyma całą wiadomość do przekazania. Eksperymenty te miały na celu określenie jak zmiana strategii wpływa na liczbę pakietów wymaganych w dekoderyze do zdekodowania całej wiadomości oraz opóźnienia dekodowania jej poszczególnych pakietów. Badania te są wstępnym krokiem w kierunku opracowania wariantu kodu fontannowego pozwalającego na strumieniową transmisję danych.

## 2. KODY RAPTOR

Kody Raptor zostały opatentowane w 2001 roku i są one najnowocześniejszym rozwiązaniem kodów fontannowych [4]. Są one rozszerzeniem kodów LT i umożliwiają odebranie danych nadawanych przy mniejszych wymaganiach. Różnią się tym, że nie wymagają one odzyskania wszystkich pakietów źródłowych, aby zdekodować całą wiadomość.

Kod Raptor składa się z kodera wstępnego oraz właściwego kodera LT. Kod wstępny (*pre-coder*) jest standardowym kodem blokowym, który ma za zadanie dodatkowo zabezpieczyć kodowaną wiadomość przed wymazaniami i pozwala on na odtworzenie części brakujących pakietów otrzymywanych na wyjściu dekodera wewnętrznego kodu LT. Zauważmy, że w przypadku kodu LT im mniejsza jest liczba jeszcze niezdekodowanych pakietów tym bardziej spada prawdopodobieństwo, że kolejny odebrany pakiet kodowy pozwoli na odtworzenie jednego z brakujących pakietów źródłowych. Oznacza, to że możliwość odtworzenia kilku ostatnich pakietów źródłowych z wykorzystaniem kodu wstępnego przekłada się na wcześniejsze zdekodowanie wiadomości. Jednocześnie pozwala to na zmniejszenie złożoności algorytmu dekodowania kodu LT poprzez użycie macierzy generującej o niższych wagach kolumn, co jednocześnie zmniejsza prawdopodobieństwo zdekodowania końcowych pakietów źródłowych.

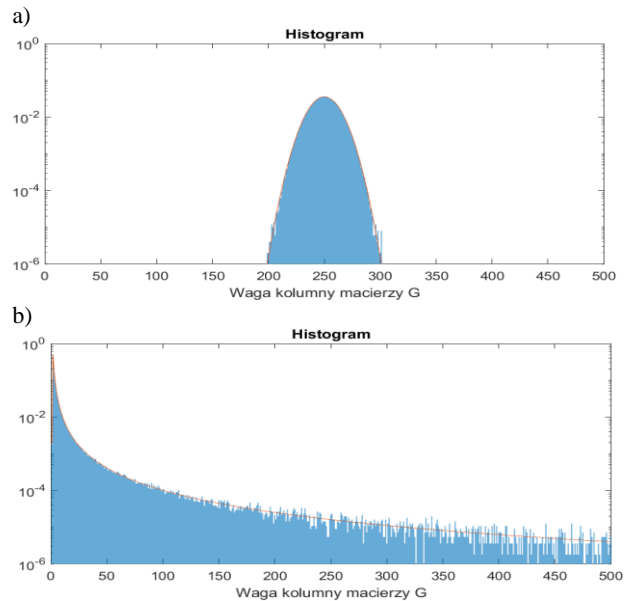
Najnowocześniejszym i najbardziej elastycznym oraz wydajnym rodzajem kodów Raptor są kody RaptorQ [7]. Kody te oferują rozwiązania spełniające wymagania zarówno użytkowników końcowych, jak również wysokiej jakości dostawców usług (QoS). Koder oraz dekoderek pozwala usługom na przesyłanie strumieniowe, odzyskanie danych utraconych w trakcie transmisji oraz ich rekonstrukcję bez użycia kanału zwrotnego. Blok źródłowy jest konfigurowany przez aplikację i jest on podzielony na równe fragmenty danych. Koder RaptorQ generuje pakiety naprawy (*repair symbols*) za pomocą pakietów źródłowych bloku. Natomiast zaszyfrowane pakiety, które są kombinacją pakietów źródłowych i naprawczych, są wysyłane do odbiorcy.

### 3. BADANIE WŁAŚCIWOŚCI WYBRANYCH WARIANTÓW KODÓW FONTANNOYCH

W ramach eksperymentów zbadano właściwości wybranych wariantów kodów fontannowych pozwalających na odzyskanie danych wymazanych w trakcie transmisji bez użycia kanału zwrotnego. Porównano tu efekty dla zastosowania dwóch rozkładów wag kolumn macierzy generującej (rys. 1). Wyniki wygenerowano na podstawie symulacji transmisji 1000 wiadomości złożonych z 500 pakietów źródłowych.

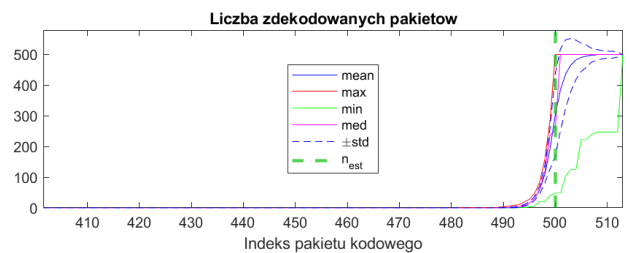
#### 3.1. Scenariusz referencyjny

W niniejszym punkcie założono transmisję przez kanał idealny a wszystkie pakiety źródłowe trafiają od razu do kodera w nadajniku, co jest podstawowym założeniem dla kodów fontannowych.



Rys. 1. Histogram wag macierzy generującej dla (a) rozkładu dwumianowego oraz (b) rozkładu solitonowego

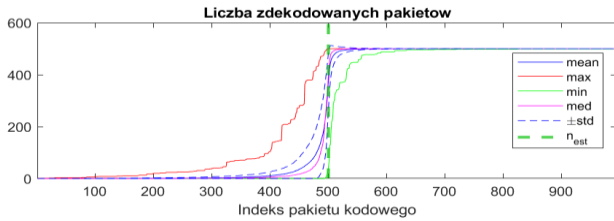
Na rys. 2 przedstawiono wyniki uzyskane dla równomiernego rozkładu jedynek w macierzy generującej  $G$ . Założono, że jedynka na danej pozycji występuje z prawdopodobieństwem 0.5, z wyłączeniem przypadku, w którym kolumna macierzy  $G$  składa się z samych zer. Przekłada się to na dwumianowy rozkład wag kolumn macierzy generującej o wartości średniej równej połowie liczby przesyłanych pakietów (rys. 1a). Dla uproszczenia dalej ten przypadek macierzy  $G$  będzie nazywany macierzą o rozkładzie dwumianowym. Zauważmy, że w tym przypadku do blisko 500 iteracji nie został zdekodowany żaden pakiet, a następnie widoczny jest gwałtowny wzrost liczby zdekodowanych pakietów i po otrzymaniu kilku dodatkowych pakietów cała wiadomość zostaje poprawnie zdekodowana.



Rys. 2. Liczba zdekodowanych pakietów w funkcji indeksu odebranego pakietu kodowego dla transmisji przez kanał idealny dla rozkładu dwumianowego

Z kolei dla solitonowego rozkładu wag kolumn macierzy generującej (rys. 3) możemy zauważyć, że pierwsze pakiety są odtwarzane po odebraniu średnio około 350 pakietów kodowych (rys. 2) chociaż, w niektórych przypadkach proces dekodowania początkowych pakietów zaczyna się już po około 100 iteracjach, Gwałtowny wzrost liczby zdekodowanych pakietów widoczny jest dla około 480 iteracji, a następnie proces ten spowolnia ponieważ maleje prawdopodobieństwo, że kolejny pakiet odebrany pozwoli na zdekodowanie kolejnego pakietu źródłowego. Cała wiadomość zostaje odebrana w połowie przypadków po około 550 pakietach kodowych, ale w najgorszym przypadku wymagane było odebranie aż 1000 pakietów kodowych.

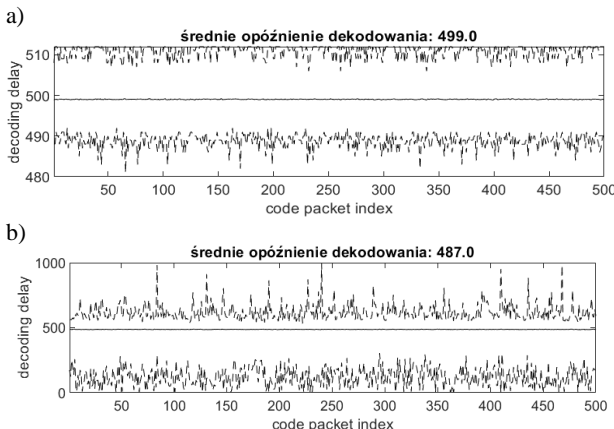
Na podstawie przedstawionych danych oraz wykresów widać, że rozkład solitonowy pomimo tego, że pozwala na wcześniejsze zdekodowanie pierwszych pakietów to równocześnie wymaga większej liczby przesłanych i odebranych pakietów zwiększając opóźnienie wynikające z kodowania.



Rys. 3. Liczba zdekodowanych pakietów w funkcji indeksu odebranego pakietu kodowego dla transmisji przez kanał idealny dla rozkładu solitonowego

Potwierdza to opóźnienie dekodowania poszczególnych pakietów źródłowych (rys. 4). Dla rozkładu o dużej wartości średniej (dwumianowego) opóźnienie dekodowania wykazuje relatywnie nieduże wahania. Minimalne zaobserwowane opóźnienie wskazuje, kiedy najwcześniej pierwszy pakiet źródłowy może dostać zdekodowany, a maksymalne zaobserwowane opóźnienie wskazuje na to, kiedy ostatni pakiet źródłowy, kończący dekodowanie całej wiadomości może zostać zdekodowany. Dla rozkładu dwumianowego (rys. 4a) widzimy, że w ramach 1000 symulowanych transmisji najwcześniej po 481 odebranych pakietach kodowych pojawił się pierwszy zdekodowany pakiet, jednocześnie ostatni pakiet został zdekodowany nie później niż po 512 pakietach kodowych.

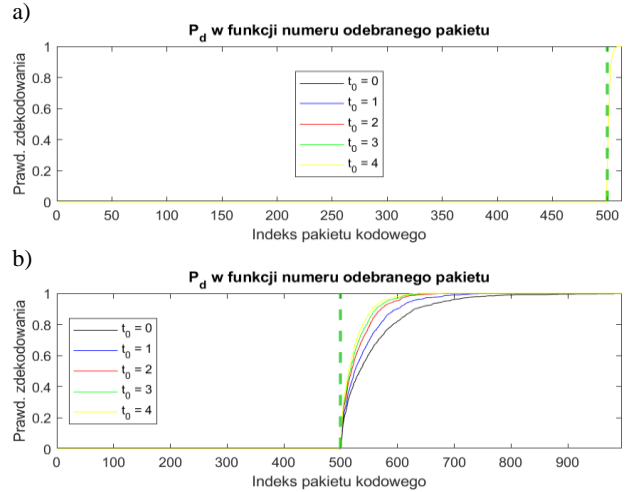
W przypadku rozkładu solitonowego (rys. 4b) chociaż obserwujemy mniejsze średnie opóźnienie dekodowania (487 zamiast 499), to jednocześnie zakres wahań opóźnienia dekodowania poszczególnych pakietów znacznie wzrasta. Wartość minimalna tym razem wynosi nawet 0 (pierwszy pakiet kodowy może być równoważny jednemu z pakietów źródłowych), ale maksymalna sięga aż 1000.



Rys. 4. Opóźnienie dekodowania pakietów źródłowych dla kanału idealnego oraz rozkładu (a) dwumianowego i (b) solitonowego; liniami przerywanymi zaznaczono wartość minimalną i maksymalną wyznaczoną na podstawie 1000 transmisji

Porównując wartości prawdopodobieństwa zdekodowania wiadomości  $P_d$  w funkcji liczby pakietów odebranych dla rozkładu dwumianowego i solitonowego (rys. 5 oraz tab. 1) możemy zauważyć, że dla rozkładu

dwumianowego bez względu na wartość  $t_0$  (oznaczającą liczbę pakietów źródłowych odtwarzanych przed koder wstępny) liczba wymaganych pakietów kodowych jest taka sama. Oznacza to, że w tym przypadku zastosowanie kodera wstępnego mija się z celem. Natomiast, dla rozkładu solitonowego wraz ze wzrostem wartości  $t_0$  potrzebna jest coraz mniejsza liczba pakietów, co uzasadnia stosowanie w tym przypadku kodera wstępnego.



Rys. 5. Prawdopodobieństwo zdekodowania dla kanału idealnego oraz rozkładu (a) dwumianowego i (b) solitonowego

Tab. 1. Liczba przesłanych pakietów kodowych wymaganych dla osiągnięcia wybranych wartości  $P_d$

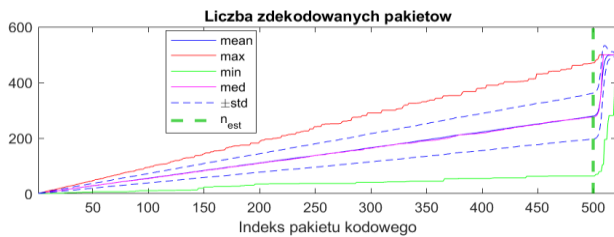
| $P_d$ | $t_0$ | rozkład dwumianowy | rozkład solitonowy |
|-------|-------|--------------------|--------------------|
| 1,0   | 0     | 513                | 990                |
|       | 1     |                    | 975                |
|       | 2     |                    | 691                |
|       | 3     |                    | 682                |
|       | 4     |                    | 680                |
| 0,95  | 0     | 505                | 666                |
|       | 1     |                    | 621                |
|       | 2     |                    | 593                |
|       | 3     |                    | 578                |
|       | 4     |                    | 570                |

### 3.2. Właściwości transmisji dla stopniowego napływania pakietów źródłowych

Rozpatrzmy teraz przypadek transmisji przez idealny kanał jednokierunkowy bez strat pakietów, w którym pakiety źródłowe stopniowo wpływają do nadajnika w trakcie przesyłania pakietów kodowych. Zakładamy tu, że pakiety kodowe generowane są w tym samym tempie jak napływają do kodera pakiety źródłowe, a maksymalna liczba pakietów źródłowych wynosi 500. Zauważmy, że zmienia to również rozkłady wag kolumn macierzy generującej, bo długość kolumny macierzy generującej rośnie wraz z przyrostem liczby pakietów źródłowych a wraz z nią rośnie średnia liczba jedynek lokowanych w kolumnie.

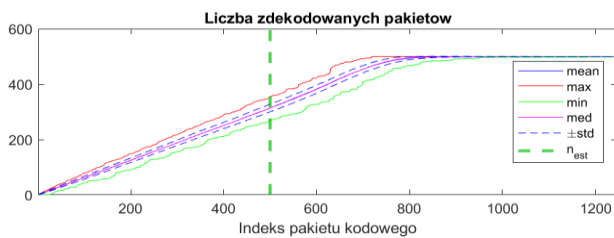
Rys. 6 przedstawia statystyki liczby zdekodowanych pakietów w funkcji indeksu pakietu kodowego pomierzone dla rozkładu dwumianowego. Jak widać tym razem pierwsze pakiety są dekodowane są w przeciągu całej transmisji i średnio na dwa otrzymane pakiety zdekodowany zostaje jeden nowy pakiet źródłowy (rys. 6). Średnio po otrzymaniu pakietów kodowych w liczbie równej maksymalnej liczbie pakietów źródłowych zostaje zdekodowana połowa pakietów. Następnie po otrzy-

maniu kilku dodatkowych pakietów liczba ta gwałtownie wzrasta i cała wiadomość zostaje zdekodowana, chociaż nieco później niż dla wcześniejszego przypadku (tab. 1 i tab. 2).



Rys. 6. Liczba zdekodowanych pakietów dla rozkładu dwumianowego i pakietów napływających do kodera stopniowo

Sytuacja przedstawia się znacznie mniej korzystnie dla rozkładu solitonowego (rys. 7), który cechują mniejsze wagi kolumn. W tym przypadku, po otrzymaniu 500 pakietów kodowych zdekodowanych zostało blisko 300 pakietów, czyli więcej niż dla rozkładu dwumianowego. Jednak tempo dekodowania nowych pakietów utrzymuje się nadal na tym samym poziomie by pod koniec jeszcze znacznie zwolnić. Wszystkie dane nadane zostają zdekodowane dopiero po odebraniu maksymalnie 1241 pakietów, gdzie dla rozkładu dwumianowego cała wiadomość zostaje zdekodowana po otrzymaniu 521 pakietów kodowych (tab. 2).



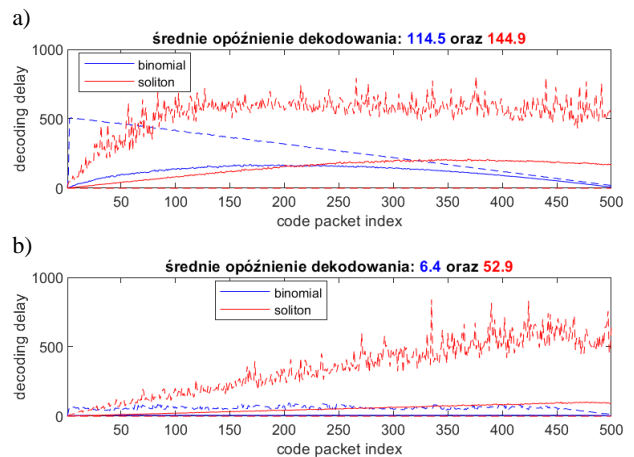
Rys. 7. Liczba zdekodowanych pakietów dla rozkładu solitonowego i pakietów napływających do kodera stopniowo

Tab.2. Liczba przesłanych pakietów kodowych wymaganych dla osiągnięcia wybranych wartości  $P_d$

| $P_d$ | $t_0$ | rozkład dwumianowy | rozkład solitonowy |
|-------|-------|--------------------|--------------------|
| 1,0   | 0     | 521                | 1241               |
|       | 1     |                    | 1203               |
|       | 2     |                    | 1177               |
|       | 3     |                    | 1125               |
|       | 4     |                    | 1045               |
| 0,95  | 0     | 512                | 1021               |
|       | 1     |                    | 954                |
|       | 2     |                    | 913                |
|       | 3     |                    | 894                |
|       | 4     |                    | 878                |

Bardzo ciekawe zjawisko można zaobserwować, jeżeli przyjmiemy, że pakiety kodowe generujemy szybciej niż napływają pakiety źródłowe. Zauważmy, że opóźnienie dekodowania dla poprzedniego przypadku (rys. 8a) cechują duże wartości średnie oraz maksymalne, malejące dla rozkładu dwumianowego dla wzrastającego numeru pakietu źródłowego. Jednak, gdy dla rozkładu dwumianowego zwiększymy stosunek liczby pakietów kodowych do liczby pakietów źródłowych do  $\alpha = 6/5$  (rys. 8b) opóźnienie dekodowania poszczególnych pakietów gwałtownie spada i pakiety źródłowe są dekodowane na bieżąco z niewielkim opóźnieniem, co umożliwia na zrealizowanie transmisji strumieniowej bez założonej z góry liczby pakietów źródłowych i jed-

nocześnie niedużej złożoności dekodowania. Rozkład solitonowy nawet dla  $\alpha = 10$  (rys. 8b) nie pozwala na uzyskanie takiego efektu.



Rys. 8. Opóźnienie dekodowania dla (a)  $\alpha = 1$  oraz (b)  $\alpha = 6/5$  (rozkład dwumianowy) oraz  $\alpha = 10$  (rozkład solitonowy)

#### 4. PODSUMOWANIE

W pracy przedstawiono wyniki badania właściwości kodu fontannego dla dwóch różnych rozkładów wag kolumn losowej macierzy generującej. Wykazano, że rozkład solitonowy o niskich wagach kolumn, chociaż zmniejsza złożoność dekodowania skutkuje znacznym zwiększeniem opóźnienia dekodowania całej wiadomości źródłowej. Jednocześnie rozkład dwumianowy o większej średniej wadze kolumn umożliwia uzyskanie efektu transmisji strumieniowej, w której kolejne pakiety źródłowe napływają do kodera stopniowo, o małym opóźnieniu dekodowania bez konieczności stosowania dodatkowego kodera wstępnego.

#### LITERATURA

- [1] Bello Philip. 1963, „Characterization of randomly time-variant channels”, IEEE Transactions on Communications Systems.
- [2] Haustein Kamila 2021. „Aplikacja demonstrująca działanie kodów fontannowych”, projekt dyplomowy inżynierski, Wydział Elektroniki, Telekomunikacji i Informatyki, Politechnika Gdańska.
- [3] Ho Tracey, 2003, „Summary of raptor codes”, Scientific Commons.
- [4] Joshi G. i inni, 2010, „Fountain codes”, Global telecommunications conference.
- [5] MacKay David JC, 2005, „Fountain codes”, IEE Proceedings-Communications, s. 1062-1068.
- [6] Nagasubramanian K., 2008: „Code design for erasure channels with limited or noisy feedback”, PhD Thesis. Texas A & M University.
- [7] Zhang K., Z. Qinyu, J. Jian, 2017, "Bounds on the reliability of RaptorQ codes in the finite-length regime", IEEE Access, 5, 24766-24774.
- [8] Chen Guo Tai, 2012, "Analysis of robust soliton distribution for LT code", IEEE 11th International Conference on Signal Processing, vol. 2, pp. 1546-1549.

