

APLIKACJA DEMONSTRUJĄCA DZIAŁANIE KODÓW FONTANNOYCH
THE APPLICATION DEMONSTRATING THE FOUNTAIN CODESKamila Hausteina¹; Marek Blok²¹ Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, Gdańsk, hausteinkamila@gmail.com² Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, Gdańsk, marek.blok@pg.edu.pl

Streszczenie: Istotną cechą kodów fontannowych jest możliwość odtwarzania nadawanych danych niezależnie od jakości kanału. Wynika to z tego, że kody te nie mają z góry założonej zawartości informacyjnej, a kolejne symbole nadmiarowe są generowane, dopóki jest taka potrzeba. Jest to szczególnie przydatne w transmisjach broadcastowych, bo każdy z odbiorców może zdekodować dane tak wcześnie, na ile pozwala jakość łącza. Aby przybliżyć funkcjonowanie tych kodów studentom, powstała zaprezentowana w pracy aplikacja demonstrująca ich kodowanie i dekodowanie.

Abstract: An important feature of the fountain codes is the ability to reconstruct the transmitted data regardless of the conditions in the channel. They are rateless codes, so they do not have any predetermined code rate, and successive redundant symbols are generated as long as it is needed. This is especially useful in broadcast transmissions, where each recipient can restore the data as early as possible depending on link quality. In order to familiarize students with the functioning of these codes, an application was created that demonstrates their coding and decoding.

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

Keywords: erasure channel, error correction coding, fountain codes.

1. WSTĘP

W 1998 roku Michael Luby zapoczątkował cyfrowe kody fontannowe LT (*Luby Transform Codes*) [6], które umożliwiają rozszerzanie kodu w trakcie transmisji o kolejne pozycje kontrolne zależnie od potrzeb. Struktura kodów nie jest z góry ustalona, stąd zawsze mamy możliwość wygenerowania kolejnego symbolu, który w zestawieniu z wybranymi poprzednimi symbolami może pozwolić na odtworzenie nadawanej wiadomości. Jednocześnie symbole nadawane można odtworzyć na wiele sposobów na podstawie różnych zestawów symboli odebranych. Wraz z dodawaniem kolejnych symboli kodowych liczba takich możliwości rośnie, a co za tym idzie rośnie odporność kodu na wymazywanie jego symboli.

Jeżeli przyjąć, że kodowane symbole to kolejne pakiety do nadania, to kody fontannowe pozwalają na odtworzenie wszystkich pakietów nadawanej wiadomości, w sytuacji, gdy kanał transmisyjny wymazuje w sposób losowy przesyłane pakiety kodowe. Ponieważ z założenia liczba zakodowanych pakietów, które może wygenerować źródło wiadomości jest nieograniczona, to niezależ-

nie od prawdopodobieństwa wymazania pakietów w kanale, odbiorca prędzej czy później odbierze zestaw pakietów wystarczający do odtworzenia całej nadawanej wiadomości.

W przypadku kodów fontannowych liczba pakietów nadawanych może być określona w trakcie kodowania zależnie od warunków panujących w kanale, bez konieczności zakładania z góry ustalonej zawartości informacyjnej kodu (*code rate*) [7]. Ta cecha kodów *rateless* jest szczególnie przydatna w przypadku transmisji rozgłoszeniowych, gdy straty pakietów dla każdego odbiorcy są inne. To odbiorca, a nie nadawca, określa, ile pakietów jest mu potrzebnych do prawidłowego odebrania wiadomości. W efekcie kod się „dopasowuje” do pojemności informacyjnej kanału. Odbiorca dysponujący kanałem wysokiej jakości odtworzy wiadomość nadawaną bez konieczności odbierania pakietów nadmiarowych lub z niewielką ich liczbą. Z kolei odbiorca z kanałem niskiej jakości będzie wymagał wielu pakietów nadmiarowych, których liczba jest powiązana z wymaganą zawartością informacyjną kodu zależną od faktycznej pojemności informacyjnej kanału, której nie musimy znać z góry. Dzięki tej właściwości kody te znalazły swoje miejsce w nowoczesnych rozwiązaniach i standardach.

W celu przybliżenia funkcjonowania tych ważnych kodów opracowana została zaprezentowana w pracy aplikacja, zrealizowana w Matlabie, demonstrująca działanie algorytmu kodowania i dekodowania kodów fontannowych [2]. Zaimplementowane narzędzie umożliwia m.in. pracę krokową, czy też obserwowanie równoczesnej transmisji do kilku odbiorców przy niezależnych stratach pakietów. Program ten umożliwia zweryfikowanie warunków koniecznych do prawidłowego odebrania nadanej wiadomości w zależności od właściwości macierzy generującej przy pakietach losowo usuwanych przez kanał podczas transmisji.

2. RODZAJE KODÓW FONTANNOYCH

Ogólną ideą kodów fontannowych jest rozpraszanie informacji o pakietach do nadania w wielu pakietach przesyłanych przez kanał w taki sposób, żeby możliwe było odtworzenie pakietów źródłowych na podstawie wielu różnych podzbiorów pakietów odebranych. Uodparnia to przesyłane dane na losowe straty pakietów. Koncepcję przypadkowych kodów fontannowych przedstawiono w p. 2.1. W ramach tej koncepcji powstały różne rozwiązania praktyczne takie jak kody Tornado [5], LT [7] oraz

Raptor [3]. Kody te i ich właściwości pokrótce przedstawiono w p. 2.2-2.4.

2.1. Przypadkowy kod fontanny

Pakiet wyjściowy kodera kodu fontannowego jest sumą modulo 2 losowych pakietów źródłowych x_l , gdzie x_l dla $l = 1, 2, \dots, N$ to l -ty pakiet, który chcemy przesłać do odbiorcy. Oznacza to, że pakiet nadawany y_j można zapisać w postaci mnożenia macierzowego realizowanego w arytmetyce modulo 2

$$y_j = \mathbf{g}_j \mathbf{x} \quad (1)$$

gdzie \mathbf{g}_j jest wektorem kolumnowym o elementach przyjmujących wartość 0 albo 1, wskazujących na pakiety źródłowe tworzące pakiet nadawany y_j . Kolejne kolumny \mathbf{g}_j tworzą macierz generującą nadajnika $\mathbf{G}_{K,N}$, gdzie K jest liczbą nadanych pakietów. Dla przypadkowego kodu fontannowego kolejne kolumny macierzy generującej $\mathbf{G}_{K,N}$ są generowane losowo.

Jak można zauważyć $\mathbf{g}_j[l] = 1$ oznacza, że w pakiecie y_j jest zawarta informacja o pakiecie źródłowym x_l . Jednak w celu odtworzenia tego pakietu z odebranego pakietu y_j należy wyeliminować wpływ pozostałych pakietów źródłowych

$$x_l = y_j \oplus \sum_{k=1}^N \mathbf{g}_j[k] x_k \quad (2)$$

Żeby odtworzyć wszystkie pakiety źródłowe x_i , odbiornik odbiera kolejne pakiety nadawane y_j tak długo, aż skompletuje N niezależnych kolumn macierzy $\mathbf{G}_{K,N}$, które tworzą macierz kwadratową \mathbf{G} . Problem sprowadza się wtedy do rozwiązania równania macierzowego

$$\mathbf{x} = \mathbf{G}^{-1} \mathbf{y} \quad (3)$$

Łatwo zauważyć, że aby przeciwdziałać stratom (wymazaniom) pakietów w kanale, wystarczy w macierzy $\mathbf{G}_{K,N}$ umieścić nadmiarowe kolumny. W takiej sytuacji zadaniem odbiornika jest wytypowanie spośród odebranych pakietów tych, które odpowiadają niezależnym kolumnom macierzy generującej.

2.2. Kody Tornado

Kody Tornado [5] w przeciwieństwie do ogólnych założeń kodów fontannowych nie należą do kodów rateless. Głównym atutem tych kodów jest możliwość przesyłania dużej liczby pakietów do wielu użytkowników zachowując przy tym małą liczbę pakietów nadmiarowych [4]. Dodatkowo cechują się one niską złożonością algorytmu dekodowania, co wynika to z tego, że pakiety nadmiarowe są zależne od niewielkiej liczby pakietów informacyjnych. Stąd kody te należą do grupy kodów LDPC (*low density parity check*) [4].

W kodzie Tornado na podstawie pakietów informacyjnych tworzona jest założona z góry liczba nadmiarowych pakietów kontrolnych. Następnie na ich podstawie tworzone są kolejne pakiety nadmiarowe chroniące pakiety kontrolne niższego poziomu. Kod taki może zawierać kilka kolejnych poziomów zabezpieczeń pozycji nadmiarowych, gdzie ostatni poziom jest zabezpieczony klasycznym kodem blokowym, który jest bardziej odporny na straty, np. kodem Reeda-Solomona [5].

Kody Tornado są oparte na rzadkim grafie dwudzielnym. Wszystkie węzły tego grafu, reprezentujące pakiety nadawane stanowią razem słowo kodowe. Wśród tych węzłów, węzły lewe reprezentują symbole informacyjne, a węzły prawe symbole kontrolne. Wielopoziomowa struktura kodów Tornado sprawia, że nie są one odpowiednie dla dużych systemów przesyłania danych ze względu na złożoność algorytmu ich kodowania oraz dekodowania proporcjonalną do długości kodu [6].

2.3. Kody LT

Kody LT (*Luby transform codes*) [7] są pierwszymi kodami fontannowymi o znaczeniu praktycznym. Charakteryzują się one tym, że liczba przesyłanych zakodowanych pakietów jest potencjalnie nieograniczona i jest określona w trakcie transmisji [7]. Zakłada się, że nadajnik przesyła kolejne zakodowane pakiety tak długo, aż dekodery będzie w stanie odtworzyć dane źródłowe.

W efekcie rozszerzania kodu na bieżąco w trakcie transmisji, dane źródłowe można odtworzyć niezależnie od liczby wymazanych pakietów. Kolejne pakiety są tworzone zgodnie z zależnością (1), przy czym wektor \mathbf{g}_j jest tworzony w koderze losowo. Najpierw losowany jest stopień bieżącego węzła kontrolnego \mathbf{d} , a następnie w wektorze \mathbf{g}_j umieszczane jest losowo \mathbf{d} jedynek. Parametr \mathbf{d} określa liczbę losowo wybranych pakietów źródłowych, które tworzą zakodowany pakiet.

Właściwościami kodu LT są ściśle związane z rozkładem prawdopodobieństwa stopni węzłów \mathbf{d} . W praktyce wykorzystuje się solidny rozkład solitonowy (*robust soliton distribution*) [5], który gwarantuje, że większość pakietów jest niskiego stopnia. Przekłada się to na niską złożoność kodowania i dekodowania oraz dużą wydajność dla dużej liczby kodowanych pakietów [8]. Ponadto kody LT są uniwersalne ze względu na to, że są one prawie optymalne dla każdego kanału z wymazywaniem.

2.4. Kody Raptor

Wadą kodów LT jest to, że wraz z kolejnymi odtworzonymi pakietami źródłowymi spada prawdopodobieństwo tego, że kolejny pakiet odebrany umożliwi odtworzenie nowego pakietu źródłowego. Kody Raptor [3] są rozszerzeniem kodów LT, które ograniczają ten problem i pozwalają na wcześniejsze odtworzenie wszystkich symboli wejściowych. Zasadniczo są one oparte na kodach LT, z tą różnicą, że kodowanie dzieli się na dwa etapy. Pakiety źródłowe są wstępnie kodowane standardowym kodem blokowym (*pre-coder*) zabezpieczającym przed wymazaniami. Następnie tak zabezpieczone pakiety są kodowane z użyciem klasycznego kodu LT. Dwustopniowa struktura kodu oznacza, że dekodery kodu LT nie musi odtworzyć wszystkich symboli kodu wstępnego, bo pozwala on na odtworzenie brakujących symboli [8]. Kosztem niewielkiej dodatkowej nadmiarowości wprowadzanej przez koder wstępny, dekodery LT nie musi oczekiwać na odebranie pakietu, do którego w ramach losowania trafiły ostatnie brakujące pakiety źródłowe [9].

Jednym z typu kodów Raptor są kody RaptorQ [3], które są najbardziej zaawansowanym kodem fontannowym. Charakteryzują się one znacznie lepszym prawdopodobieństwem dekodowania i lepszą wydajnością w porównaniu z użyciem samego kodu LT.



3. DEKODOWANIE KODÓW FONTANNOWYCH

W poprzednim punkcie opisano koncepcję kodowania kodów fontannowych, z kolei w niniejszym punkcie przedstawiono opis algorytmu dekodowania, który został zaimplementowany w aplikacji (rys. 2). Algorytm ten na podstawie odebranych pakietów i odpowiadających im kolumn macierzy G tworzy macierz \hat{G} składającą się z zestawu N jej niezależnych kolumn oraz odpowiadający jej zestaw pakietów \hat{Y} . Jest wystarczające do odtworzenia wektora x (3). Równocześnie, już w trakcie tej selekcji, rozwiązywany jest układ równań (3). Algorytm składa się z sześciu głównych etapów:

1. Inicjalizacja dekodera.
2. Odbiór kolejnego pakietu y_{k+1} i generacja odpowiadającej mu kolumny g_{k+1} .
3. Wstępne przetworzenie pakietu y_{k+1} oraz kolumny g_{k+1} z użyciem bieżącej wersji macierzy \hat{G} oraz wektora \hat{Y} w celu wyzerowania elementów g_{k+1} na pozycjach od 1 do k .
4. Weryfikacja czy przetworzona kolumna g_{k+1} może być dołączona do macierzy \hat{G} . (4a) Jeżeli przetworzona kolumna g_{k+1} ma jedynkę na $k + 1$ pozycji, to dołączenie jej do macierzy \hat{G} oraz dołączenie pakietu y_{k+1} do wektora \hat{Y} i aktualizacja macierzy \hat{G}_R oraz wektora \hat{Y}_R . (4b) Jeżeli przetworzona kolumna g_{k+1} ma zero na $k + 1$ pozycji, to dołączenie jej do macierzy rezerwowej \hat{G}_R oraz dołączenie pakietu y_{k+1} do wektora \hat{Y}_R .
5. Sprawdzenie czy odtworzono wszystkie pakiety nadawanej wiadomości x . Jeżeli $k = N + 1$ kończymy dekodowanie (7), w przeciwnym razie przechodzimy do przetwarzania kolejnego pakietu: zwiększenie k o jeden.
6. Weryfikacja czy w macierzy rezerwowej \hat{G}_R są kolumny z jedynką na $k + 1$ pozycji. Jeżeli tak, to (7a) wydobycie pierwszej z nich jako kolumny g_{k+1} oraz z wektora \hat{Y}_R i przejście do p. 4a. Jeżeli nie, to przejście do p. 2.

4. APLIKACJA DEMONSTRUJĄCA

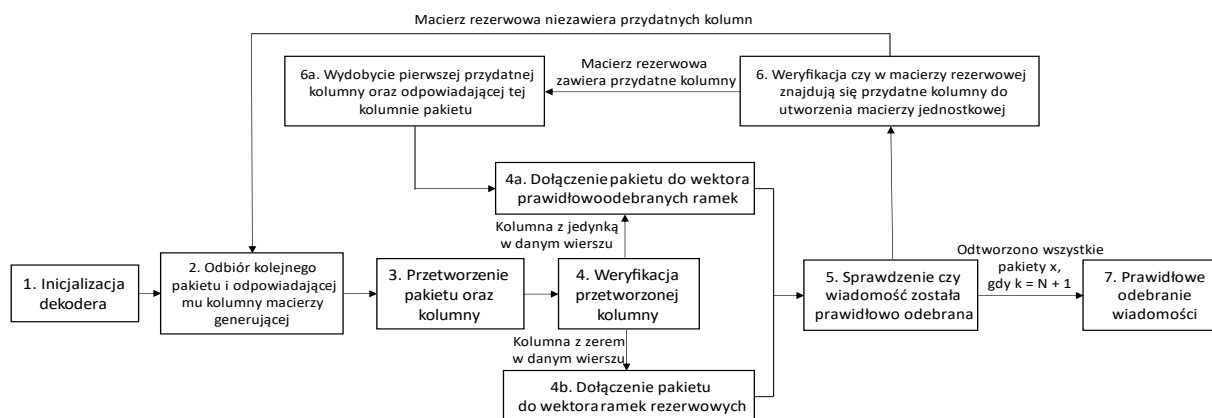
Opracowana aplikacja demonstruje działanie kodów fontannowych. Została ona zorganizowana w sposób, który jest intuicyjny dla użytkownika i łatwy w obsłudze.

Umożliwia ona wybór symulacji transmisji do jednego odbiorcy, kilku odbiorców oraz testowanie kodów fontannowych. Użytkownik samodzielnie może dokonać wyboru wartości parametrów takich jak: typ danych nadawanych, ich długość, liczbę pakietów, typ macierzy generującej, prawdopodobieństwo wymazywania lub numery usuniętych pakietów. W aplikacji zostały przedstawione również efekty transmisji w postaci parametrów liczbowych, takich jak liczba iteracji, liczba odebranych pakietów, liczba wymazanych pakietów, łączny czas dekodowania, najdłuższy czas dekodowania jednego pakietu itd. Dodatkowo wyniki zostały zaprezentowane w postaci wykresów, które ułatwiają zrozumienie algorytmu kodowania i dekodowania.

4.1. Przesyłanie danych do jednego odbiorcy

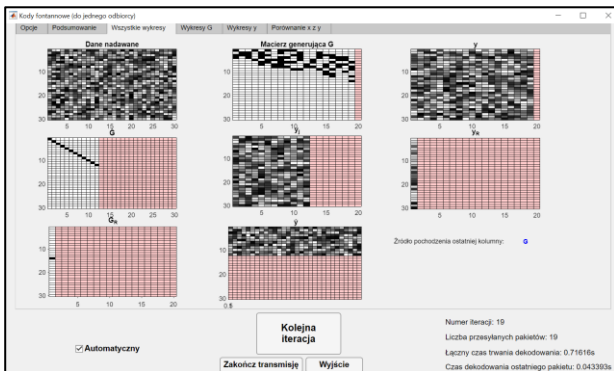
Po wyborze tej opcji użytkownik ma możliwość wyboru typu danych nadawanych: losowe lub wskazany obraz. Jeżeli wskazany zostanie obraz na ekranie będzie wyświetlona informacja odnośnie jego nazwy oraz rozmiaru. Kolejnym wyborem jest wybór sposobu generowania macierzy generującej. Zależnie od tego wyboru możliwe jest obserwowanie różnic w przebiegu procesu dekodowania dla różnych strategii generowania macierzy generującej. Mamy tu sześć możliwości.

Losowy: Wykorzystywane kolumny macierzy generującej są generowane w sposób losowy, a odpowiadające im pakiety odbierane, aż do momentu prawidłowego odebrania wiadomości. Scenariusz 1: Macierz generująca jest macierzą jednostkową, która ma wymiary równe liczbie pakietów danych nadawanych. Scenariusz 2: Kolumny macierzy generującej są generowane losowo. Dekodowanie zatrzymuje się po odebraniu N pakietów. W przypadku nieprawidłowego odebrania wiadomości można kontynuować dekodowanie z losową macierzą generującą. Scenariusz 3: Generowana jest macierz, która uniemożliwia prawidłowe odebranie wiadomości bez skorzystania z możliwości wygenerowania dodatkowych kolumn. W celu prawidłowego odebrania wiadomości użytkownik musi kontynuować odbieranie kolejnych pakietów. Scenariusz 4: Umożliwia wczytanie macierzy generującej z pliku tekstowego. Scenariusz 5: Macierz generująca posiada jedynki, które są skupione lokalnie, pozwalające na odtworzenie pakietów na podstawie bliskich sobie pakietów odebranych. Użytkownik określa zawartość informacyjną kodu oraz wielkość okna pakietów źródłowych składających się na pakiet kodowy. Okno aplikacji



Rys. 1. Schemat algorytmu dekodowania kodu fontannowego

podczas procesu dekodowania losowych danych źródłowych po wyborze Scenariusza 5 zostało zaprezentowane na rys. 2.



Rys. 2. Okno programu po wyborze scenariusza 5

4.2. Przesyłanie danych do kilku odbiorców

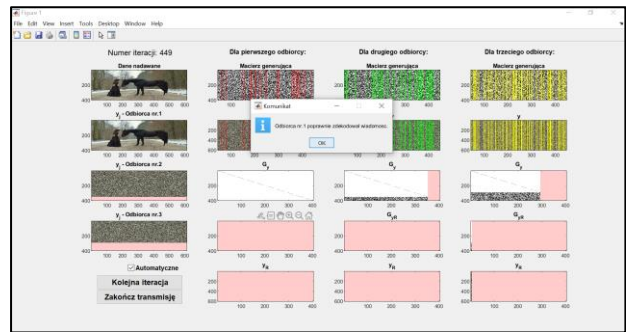
Zaimplementowane narzędzie pozwala użytkownikowi na śledzenie i przesyłanie jednakowej wiadomości do trzech różnych odbiorców i równoległe porównywanie kolejnych etapów procesu dekodowania. W celu łatwiejszego zobrazowania algorytmu działania, każdemu odbiorcy przypisano unikatowy kolor wymazywanych pakietów co możemy zauważyć na rys. 3. Po poprawnym odebraniu wiadomości przez wszystkich trzech odbiorców otwiera się zakładka Podsumowanie. W ramach tej zakładki użytkownik ma możliwość porównania wartości parametrów opisujących efektywność dekodowania przesyłanej wiadomości w zależności od zadanego prawdopodobieństwa wymazywania.

4.3. Testowanie kodów fontannowych

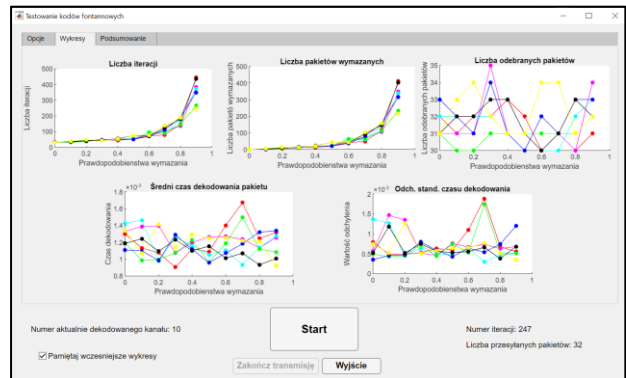
Moduł ten umożliwi użytkownikowi zaobserwowanie zmian w skuteczności procesu dekodowania w zależności od wartości prawdopodobieństwa wymazywania dla różnych macierzy generujących lub odmiennego wzorca wymazania pakietów. Użytkownik ma możliwość zachowania wcześniejszych wyników i dołożenie do nich wyników testów dekodowania, dla kolejnych parametrów kodera lub typów macierzy generującej, wykreślonych w funkcji prawdopodobieństwa wymazywania (rys. 4).

5. PODSUMOWANIE

Opracowana aplikacja umożliwia łatwiejsze zrozumienie działania kodów fontannowych. Dodatkowo ukazuje ich właściwości i zastosowania. Użytkownik ma możliwość dokładnego przeanalizowania algorytmu kodowania i dekodowania obserwując między innymi, że kody te pozwalają na prawidłowe odtworzenie przesyłanych danych niezależnie od prawdopodobieństwa wymazywania pakietów, które wpływa jedynie na to jak szybko nastąpi prawidłowe odtworzenie wiadomości. Dalsze prace nad aplikacją będą ukierunkowane na uzupełnienie jej o możliwość wyboru konkretnych typów kodów fontannowych (Tornado, LT czy Raptor), co pozwolił na porównywanie skuteczność poszczególnych rozwiązań oraz ocenić różnice w sposobie ich działania i porównanie prawdopodobieństwa poprawnego odebrania wiadomości dla określonej liczby zakodowanych pakietów przesyłanych przez kanał.



Rys. 3. Transmisja do wielu odbiorców; prawidłowe odtworzenie wiadomości przez pierwszego odbiorcę



Rys. 4. Okno aplikacji z wykresami przedstawiającymi wyniki badania wpływu prawdopodobieństwa wymazywania na poszczególne mierzone parametry symulowanych transmisji

LITERATURA

- [1] Bello Philip. 1963, „Characterization of randomly time-variant channels”, IEEE Transactions on Communications Systems, 11 (4): 360–393.
- [2] Haustein K. 2021. „Aplikacja demonstrująca działanie kodów fontannowych”, projekt dyplomowy inżynierski, Wydział Elektroniki, Telekomunikacji i Informatyki, Politechnika Gdańska.
- [3] Ho T., 2003, „Summary of raptor codes”, Scientific Commons.
- [4] Joshi G. i inni, 2010, „Fountain codes”, Global telecommunications conference.
- [5] Khisti A., 2003, „Tornado codes and Luby transform codes”, s. 1-12.
- [6] Luby M. G., Mitzenmacher M., Shokrollahi M. A., Spielman D. A., Stemann V, 1997, „Practical lossresilient codes”, s. 150-159.
- [7] MacKay, David JC, 2003: „Information theory, inference and learning algorithms”.
- [8] MacKay, David JC, 2005, „Fountain codes”, IEE Proceedings-Communications, s. 1062-1068.
- [9] Nagasubramanian K., 2008: „Code design for erasure channels with limited or noisy feedback”.
- [10] Yao W. i inni, 2008, „Research on fountain codes in deep space communication”, Congress on Image and Signal Processing.