Sieci komputerowe
Lista 2

Mateusz Markiewicz

1 czerwca 2020

1 Zadanie 1

$\tau = \frac{2{,}5km}{10^8 \frac{m}{s}} = \frac{2500m}{10^8 \frac{m}{s}} = 2.5 * 10^{-5}s$
Czas wysłania ramki $\geq 2 * \tau$
Czas wysłania ramki $= \frac{\text{wielkość ramki}}{\text{prędkość przesyłania}} = \frac{\text{wielkość ramki}}{10^7 \frac{b}{s}}$
$\frac{\text{wielkość ramki}}{10^7 \frac{b}{s}} \geq 2 * \tau$
$\frac{\text{wielkość ramki}}{10^7 \frac{b}{s}} \geq 2 * 2.5 * 10^{-5}s$
$\frac{\text{wielkość ramki}}{10^7 \frac{b}{s}} \geq 5 * 10^{-5}s$
wielkość ramki $\geq 5 * 10^{-5}s * 10^7 \frac{b}{s}$
wielkość ramki $\geq 500b$
Stąd minimalna wielkość ramki to 64 bajty.

2 Zadanie 2

W każdej rundzie każdy z n uczestników chce wysłać ramkę i robi to z prawdopodobieństwem p. $I - ty$ uczestnik wyśle z powodzeniem swoją ramkę, jeśli w danej rundzie był jedynym, który próbował coś wysłać.
Niech $P(p, n)$ oznacza prawdopodobieństwo, że dowolnemu uczestnikowi udało się wysłać ramkę w danej turze. Wiemy, że:
$P(p, n) = n * p * (1 - p)^{n-1}$
ponieważ jest n uczestników, żeby danemu z nich się udało musi wysłać swoją wiadomość (p) oraz żaden z pozostałych uczestników nie może nic wysyłać ($(1 - p)^{n-1}$).

Chcemy wyznaczyć p maksymalizujące $P(p,n)$ dla ustalonego n.

$$\begin{aligned}\max_p P(p,n) &= \max_p n*p*(1-p)^{n-1} \\ &= \max_p \log\left(n*p*(1-p)^{n-1}\right) \\ &= \max_p \log(n)+\log(p)+(n-1)\log(1-p)\end{aligned} \tag{1}$$

Niech:

$$F(p) = \log(n)+\log(p)+(n-1)\log(1-p) \tag{2}$$

$$\begin{aligned}F^{'}(p) &= \frac{1}{p} - \frac{n-1}{1-p} \\ F^{'}(p) &= 0 \\ \frac{1}{p} - \frac{n-1}{1-p} &= 0 \\ \frac{1}{p} &= \frac{n-1}{1-p} \\ \frac{1-p}{p} &= n-1 \\ \frac{1}{p} - 1 &= n-1 \\ p &= \frac{1}{n}\end{aligned} \tag{3}$$

$$\begin{aligned}\lim_{n\to\infty} P(\frac{1}{n},n) &= \lim_{n\to\infty} n*\frac{1}{n}*(1-\frac{1}{n})^{n-1} \\ &= \lim_{n\to\infty}(1-\frac{1}{n})^{n-1} \\ &= \lim_{n\to\infty}(1-\frac{1}{n})^{n} \\ &= \frac{1}{e}\end{aligned} \tag{4}$$

3 Zadanie 3

Ethernet capture effect to sytuacja, w której jedna ze stacji nadaje znacznie częściej, pomimo że wszystkie stacje stosują algorytm CSMA/CD. Zjawisko to można łatwo zobrazować mając dwie stacje - A,B. Obie stacje chcą coś nadać i zaczynają to robić w tym samym czasie - stąd powodują kolizje. Obie stacje losują wartość ze zbioru $\{0,1\}$ i odczekują tyle tur. Zakładając, że A wylosowała mniejszą wartość, niż B to A zacznie nadawać szybciej i B będzie musiał poczekać (w tym momencie sytuacja jest w pełni symetryczna). Zakładamy, że A po zakończeniu nadawania chce nadawać ponownie, B ponownie próbuje wysłać pierwszą wiadomość. Ponownie mamy kolizję, ale dla B jest to druga kolizja pod rząd, stąd A losuje wartość ze zbioru $\{0, 1\}$, a B ze zbioru $\{0, 1, 2, 3\}$. Tym razem prawdopodobieństwo, że A wylosuje mniejszą wartość jest już dużo większe, stąd całkiem prawdopodobnie to ponownie A zacznie nadawać pierwsze. Jeśli A ma nadal coś do nadania ponownie zdarzy się kolizja, A będzie losowało wartość ze zbioru $\{0, 1\}$, a B ze zbioru $\{0, 1, ..., 6, 7\}$, stąd jest bardzo prawdopodobne, że to A ponownie będzie nadawać, sytuacja będzie się powtarzała. Po kilku turach prawdopodobieństwo, że to B będzie musiało odczekać mniejszą liczbę rund i przerwie serię kolizji staje się bardzo małe.

4 Zadanie 4

Wiadomość $m = 1010$, stąd $M(x) = x^3 + x$.
Mając dane $G(x)$ oraz $r = st(G)$ chcemy znaleźć sumę kontrolną (x) t.że $G(x)|x^r M(x) + S(x)$.
Z wykładu wiemy, że $S(x) = R(x)$, gdzie $R(x)$ to reszta z cielenia $x^r M(x)$ przez $G(x)$.

1)

$G(x) = x^2 + x + 1$ oraz $r = 2$

$$x^2 M(x) = G(x) * (x^3 + x^2 + x) + x$$

Stąd $S(x) = R(x) = x$, czyli $s = 10$.

2)

$G(x) = x^7 + 1$ oraz $r = 7$

$$x^7 M(x) = G(x) * (x^3 + x) + x^3 + x$$

Stąd $S(x) = R(x) = x^3 + x$, czyli $s = 0001010$.

5 Zadanie 5

$G(x) = x + 1$ oraz $r = 1$.
Wiemy, że $st(s) \leq r - 1$, stąd $st(S) = 0$. Wiemy więc, że wartość $S(x)$ nie jest zależna od x, nazwijmy tą wartość z (oczywiście $z \in 0, 1)$.

Weźmy dowolną wiadomość $m = m_k m_{k-1} ... m_1 m_0$, wówczas
$M(x) = m_k x^k + m_{k-1} x^{k-1} + ... + m_1 x^1 + m_0$
$xM(x) = m_k x^{k+1} + m_{k-1} x^k + ... + m_1 x^2 + m_0 x$

Bit parzystości wiadomości sprawia, że cała wiadomość, razem z tym bitem ma parzystą liczbę zapalonych bitów. Bit ten musi być więc równy 1, jeśli liczba zapalonych bitów w wiadomości jest nieparzysta i 0 w przeciwnym przypadku. Można więc łatwo zaobserwować, że wartość bitu parzystości to wartość $M(1)$ w przestrzeni modulo 2. Wystarczy więc, że pokażemy, że $M(1) \equiv_{mod2} z$.

$$\begin{aligned} xM(x) &= G(x) * Q(x) + R(x) \\ &= G(x) * Q(x) + S(x) \\ &= G(x) * Q(x) + z \\ &= (x+1) * Q(x) + z \\ \\ 1M(1) &= (1+1) * Q(1) + z \\ \\ M(1) &\equiv_{mod2} z \end{aligned} \tag{5}$$

6 Zadanie 7

Odległość Hamminga dwóch kodów = minimalna liczba bitów, które musimy zmienić, żeby zmienić jeden kod w drugi (z prezentacji).
Jeśli mamy kodowanie gwarantujące, że odległość Hamminga między dowolną parą kodów to co najmniej k:

- potrafimy wykryć do k-1 błędów pojedynczych bitów,
- potrafimy skorygować do (k-1)/2 błędów pojedynczych bitów.

W kodowaniu Hamminga(7,4) do 4 bitów danych dodajemy 3 bity parzystości, każdy dla innych 3 bitów danych, a dokładniej:

- $p_1 = d_1 + d_2 + d_4$
- $p_2 = d_1 + d_3 + d_4$
- $p_3 = d_2 + d_3 + d_4$

Teraz pokażemy, że kodowanie Hamminga(7,4) dla dwóch dowolnych różnych ciągów danych ($d_1 d_2 d_3 d_4$ oraz $d'_1 d'_2 d'_3 d'_4$) różnią się na minimum 3 bitach.

- Jeśli ciągi danych różnią się na 3 lub 4 pozycjach własność zachodzi
- Jeśli ciągi danych różnią się na 2 bitach wówczas zawsze istnieje przynajmniej jeden bit parzystości w którym uwzględniamy dokładnie jeden z tych 2 różniących się bitów danych, więc własność zachodzi
- Jeśli ciągi danych różnią się na 1 bicie danych wówczas zawsze istnieją przynajmniej 2 bity parzystości w których uwzględniamy ten bit danych, więc własność zachodzi

Korzystając z wiedzy z wykładu wiemy, że jeśli odległość Hamminga między dowolną parą kodów wynosi co najmniej $k = 3$ wówczas potrafimy skorygować do $(k-1)/2 = 1$ błędów pojedynczych bitów.

7 Zadanie 9

Mamy 4-bitową wiadomość $m = m_3m_2m_1m_0$, $G(x) = x^3 + x + 1$ oraz $r = 3$.
$M(x) = m_3x^3 + m_2x^2 + m_1x + m_0$
$x^3M(x) = m_3x^6 + m_2x^5 + m_1x^4 + m_0x^3$

Wyznaczmy $S(x) = R(x)$
$x^3M(x) = G(x) * [m_3x^3 + b_2x^2 + (b_1 + b_3)x + (b_0 + b_2 + b_3)] + [(b_1 + b_2 + b_3)x^2 + (b_0 + b_1 + b_2)x + (b_0 + b_2 + b_3)]$

Stąd $s = p_2p_1p_0$, gdzie:

- $p_2 = b_1 + b_2 + b_3$
- $p_1 = b_0 + b_1 + b_2$
- $p_0 = b_0 + b_2 + b_3$

Możemy zauważyć podobieństwo do kodowania Hamminga(7,4). W analogiczny sposób, jak w zadaniu poprzednim można udowodnić, że odległość między dwoma poprawnymi kodami jest nie mniejsza niż 3.

Pokażemy teraz w jaki sposób możemy wykryć przekłamanie na jednym bicie. Weźmy kod K' który powstał poprzez przekłamanie dowolnego poprawnego kodu K na dowolnym bicie. Pokażemy najpierw, że poprzez zmianę jednego dowolnego bitu w kodzie K' możemy uzyskać jedynie jeden poprawny kod. Załóżmy nie-wprost, że zamieniając $i - ty$ bit K' uzyskaliśmy poprawny kod K_i, a zmieniając $j - ty$ (gdzie $i \neq j$) bit K' uzyskaliśmy poprawny kod K_j. Niech $dist_H(K, K')$ oznacza odległość Hemminga kodów K oraz K'. Wiemy, że dla dowolnych dwóch poprawnych kodów ta odległość jest nie mniejsza niż 3. Wiemy również, że w jednym kroku z kodu K możemy uzyskać zarówno K_i, jak i K_j, stąd $dist_H(K_i, K_j) = 2$, stąd jeden z kodów K_i lub K_j musi być niepoprawny. Stąd jeśli z kodu K' poprzez zmianę jednego bitu uzyskamy poprawny kod wiemy, że jest to K. Stąd próbując zmienić każdy z 7 bitów wiadomości możemy z kodu K' otrzymać poprawny kod K.