

Rozdział 12

Ocena efektywności protokołu AX.25

Bartłomiej ZIELIŃSKI
Politechnika Śląska, Instytut Informatyki
Bartlomiej.Zielinski@polsl.pl

Streszczenie

Przedyskutowano podstawowe właściwości protokołu AX.25, używanego jako warstwa liniowa m. in. w amatorskiej sieci Packet Radio. Przedstawiono analityczne zależności, pozwalające oszacować wydajność protokołu. Zaprezentowano wykresy, przedstawiające wpływ poszczególnych parametrów transmisji na teoretyczne osiągi protokołu.

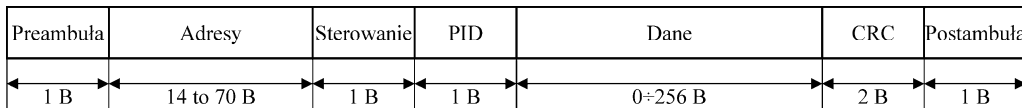
1. Wprowadzenie

Protokół AX.25 jest używany między innymi w amatorskiej sieci Packet Radio jako warstwa liniowa. Protokół ten jest zmodyfikowaną wersją szeroko znanego protokołu HDLC [1]. Warto przy tym zauważyć, że w protokole AX.25 zawarto także pewne mechanizmy typowe dla warstwy sieciowej (kierowanie ramek poprzez określone przez operatora stacje pośredniczące) oraz transportowej (sterowanie przepływem).

Sieć Packet Radio można rozpatrywać jako przykład prostej bezprzewodowej sieci rozległej. Wprawdzie – jako dzieło radioamatorów – nigdy nie było to rozwiązanie popularne (o czym świadczyć może niewielka ilość literatury poruszającej ten temat), a obecnie, wskutek powszechnej dostępności Internetu raczej trudno spodziewać się dalszego jej rozwoju, jednak można wykorzystać pewne jej elementy w procesie dydaktycznym, prezentując studentom wybrane aspekty projektowania sieci bezprzewodowych oraz urządzeń służących do transmisji w tych sieciach.

2. Protokół AX.25

Jak wspomniano powyżej, protokół AX.25 jest zmodyfikowaną wersją protokołu HDLC. W protokole tym używa się większości typów ramek protokołu HDLC; również zasady wymiany ramek są te same [2]. Modyfikacje obejmują elementy charakterystyczne dla sieci Packet Radio i uwzględniają specyfikę łączności radioamatorskiej. Ogólny format ramki protokołu AX.25 przedstawiono na rys. 1.



Rys. 1. Ogólny format ramki protokołu AX.25

Pole adresowe zawiera adresy od 2 do 10 stacji. Oprócz adresu nadawcy i odbiorcy ramki może się tu zmieścić do 8 adresów stacji pośredniczących. Każdy adres stacji zawiera do 7 bajtów, co wynika z zastosowania tzw. „znaków wywoławczych” nadawanych radioamatorom po uzyskaniu licencji. Najkrótsze pole adresowe zawiera zatem 14 bajtów, najdłuższe zaś – 70.

Pole sterujące określa między innymi typ ramki. Podobnie jak w HDLC, wyróżnia się ramki informacyjne (I, ang. *information*), nienumerowane (U, ang. *unnumbered*) oraz sterujące (C, ang. *control*). Pierwsze dwa typy ramek wykorzystuje się do przenoszenia danych, przy czym ramki informacyjne po nawiązaniu połączenia logicznego, nienumerowane zaś – w pozostałych przypadkach. Warto zauważyć przy tym, że tylko ramki informacyjne są potwierdzane.

Pole danych, występujące jedynie w ramach informacyjnych i nienumerowanych, może zawierać do 256 bajtów. Rozmiar ten można dalej ograniczać za pomocą parametru *PacLen* w zależności od aktualnej jakości łącza lub innych czynników.

Suma kontrolna CRC obliczana jest zgodnie z algorytmem CRC-CCITT. Zależnie od tego, czy suma odebrana jest zgodna z obliczoną na podstawie zawartości odebranej ramki, odbiorca wysyła potwierdzenie pozytywne lub negatywne. Podobnie jak w protokole HDLC, przed wysłaniem potwierdzenia można przesłać do 7 ramek; wielkość tę (tzw. rozmiar okna) można dalej ograniczyć za pomocą parametru *MaxFrame* w zależności np. od jakości łącza. Liczba ramek wysłanych nie zawsze jest równa tej wartości – zależy to od formatu przesyłanych danych oraz możliwości sprzętu transmisyjnego.

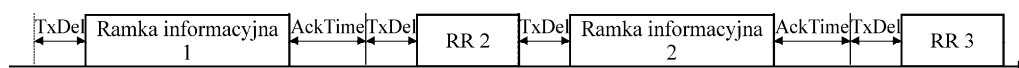
W celu zachowania przezroczystości protokołu, którą można naruszyć przez używanie specjalnych sekwencji bitów w roli preambuły i postambuły, stosuje się mechanizm zwany „szpikowaniem zerami” (ang. *bit stuffing*), a obejmujący całą ramkę z wyjątkiem preambuły i postambuły. Po każdej grupie pięciu bitów o wartości „1” automatycznie wstawia się bit „0”. Operacja ta jest odwracana w odbiorniku.

W przeciwieństwie do HDLC, protokół AX.25 wykorzystuje zrównoważony tryb asynchroniczny (ABM, ang. *Asynchronous Balanced Mode*), w którym wszystkie stacje są sobie równe. W tym trybie nadawca wysyła ramki informacyjne, które są potwierdzane przez odbiorcę. Potwierdzenie może być przy tym pozytywne (RR, ang. *Receiver Ready*) lub negatywne (REJ, ang. *Reject*). Protokół posiada także pewne elementy sterowania przepływem, które można wykorzystać, gdy odbiorca nie jest w stanie odebrać większej ilości danych, ale ostatnia ramka była odebrana bezbłędnie – odbiorca wysyła wtedy ramkę RNR (ang. *Receiver Not Ready*).

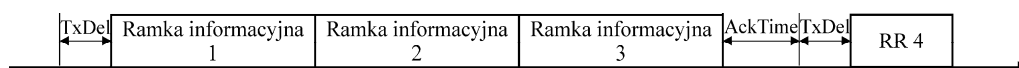
W protokole AX.25 wprowadzono dodatkowe parametry, określające właściwości łącza radiowego. Należą do nich, między innymi, *TxDel* oraz *AckTime*. Pierwszy z nich określa odstęp czasowy między włączeniem nadajnika a rozpoczęciem transmisji danych. Czas ten jest potrzebny na stabilizację parametrów nadajnika i wykrycie nośnej oraz synchronizację po stronie odbiorczej. W zależności od użytego sprzętu

czas ten może wynosić od kilku do kilkuset milisekund. Parametr *AckTime* z kolei określa, jak długo odbiorca powinien czekać przed wysłaniem potwierdzenia. Czas ten zależy od wykorzystywanej prędkości transmisji i w wielu przypadkach jest ustawiany automatycznie przez urządzenia transmisyjne. Parametr ten często nie jest potrzebny, ponieważ większość urządzeń nadawczo-odbiorczych potrafi wymusić potwierdzenie, ustawiając bit sterujący Poll/Final w ostatniej ramce informacyjnej w ciągu; w takim przypadku odbiorca nie czeka na kolejne ramki i wysyła potwierdzenie natychmiast.

Przykładowe zasady wymiany ramek dla różnych wielkości okna (parametr *MaxFrame*) pokazano na rys. 2 i 3.



Rys. 2. Wymiana ramek przy rozmiarze okna równym 1



Rys. 3. Wymiana ramek przy rozmiarze okna równym 3

3. Ocena wydajności protokołu AX.25

Dla potrzeb oceny wydajności protokołu założmy, że komunikacja odbywa się w warunkach idealnych, pozwalają one bowiem na uzyskanie najwyższych osiągnięć. Przyjmijmy zatem następujące założenia:

- sieć składa się z dwóch stacji – łącze jest przydzielone tylko jednej transmisji;
- nie występują kolizje ani błędy transmisji – nie ma potrzeby retransmisji;
- czas przetwarzania ramek jest pomijalny – nie występują dodatkowe opóźnienia.

3.1. Czas przesyłu pojedynczych ramek

Dla celów analizy wydajności protokołu założmy, że ramka sterująca ma długość 20 bajtów. Rozmiar ten odnosi się do ramki potwierdzenia; ramka informacyjna zawiera dodatkowo nie więcej niż *PacLen* bajtów danych. Ze względu na szpikowanie zerami, używane dla uzyskania przezroczystości protokołu, długość ramki dodatkowo zwiększa się, średnio o 1/63 [3]. Zatem, biorąc pod uwagę, że przed faktycznym rozpoczęciem transmisji musi upłynąć czas *TxDel*, czasy transmisji ramki danych i potwierdzenia można wyrazić następująco:

$$T_{Data} = TxDel + \frac{64}{63} \cdot \frac{160 + 8 \cdot PacLen}{R_{wl}} \quad (1)$$

oraz

$$T_{Ack} = AckTime + TxDel + \frac{64}{63} \cdot \frac{160}{R_{wl}}, \quad (2)$$

gdzie R_{wl} oznacza prędkość transmisji łącza radiowego, natomiast *AckTime* oznacza czas upływający przed odesłaniem potwierdzenia przez odbiorcę.

3.2. Łącze dwukierunkowe naprzemiennie

W przypadku łącza dwukierunkowego naprzemiennego (ang. *half-duplex*), gdy stacje wymieniają tylko niewielkie porcje danych (nie większe niż $PacLen$), ramki danych i potwierdzenia przesyłane są naprzemiennie, jak pokazano na rys. 2. Podczas przesyłu większej ilości danych nadawca może wysłać kolejno więcej ramek, nie przekraczając jednak ustalonej wielkości okna ($MaxFrame$). Ramki te zostaną potwierdzone wspólnie, jak pokazano na rys. 3.

3.2.1. Całkowity czas transmisji informacji

Zakładając, iż całkowity rozmiar danych wynosi $PacLen \times MaxFrame$, długość cyklu transmisyjnego można wyrazić jako:

$$T_p = TxDel + MaxFrame \cdot \left(\frac{64}{63} \cdot \frac{160 + 8 \cdot PacLen}{R_{wl}} \right) +$$

$$+ AckTime + TxDel + \frac{64}{63} \cdot \frac{160}{R_{wl}} \quad (3)$$

lub, w postaci uproszczonej

$$T_p = AckTime + 2 \cdot TxDel + (1 + MaxFrame) \cdot \frac{64}{63} \cdot \frac{160}{R_{wl}} +$$

$$+ MaxFrame \cdot \left(\frac{64}{63} \cdot \frac{8 \cdot PacLen}{R_{wl}} \right). \quad (4)$$

Warto przy tym zauważyć, że pierwszy składnik równania (4) określa wyłącznie narzut protokołu, podczas gdy czas niezbędny do przesyłu samych danych określony jest przez drugi składnik równania.

Gdy całkowity rozmiar danych jest różny od iloczynu $PacLen \times MaxFrame$, ostatni cykl transmisyjny będzie krótszy, niż pozwalają na to ustawione wartości tych parametrów. Można zatem zauważyć niewielką utratę wydajności, ponieważ końcowy fragment danych będzie przesłany z większym narzutem protokołu (mniej bitów danych na tę samą liczbę bitów sterujących). Dla całego pliku o rozmiarze $FileSize$ bajtów liczbę cykli transmisyjnych można określić jako:

$$n = \left\lceil \frac{FileSize}{PacLen \cdot MaxFrame} \right\rceil. \quad (5)$$

Przesył całego pliku będzie zatem zajmował czas

$$T_p = \left\lceil \frac{FileSize}{PacLen \cdot MaxFrame} \right\rceil \cdot \left(AckTime + 2 \cdot TxDel + (1 + MaxFrame) \cdot \frac{64}{63} \cdot \frac{160}{R_{wl}} \right) +$$

$$+ \left\lceil \frac{FileSize}{PacLen} \right\rceil \cdot \left(\frac{64}{63} \cdot \frac{8 \cdot PacLen}{R_{wl}} \right). \quad (6)$$

Podobnie jak poprzednio, pierwszy składnik równania oznacza narzut protokołu, drugi zaś – czas potrzebny na przesłanie samych bitów danych.

3.2.2. Wydajność protokołu

Dla dowolnego zestawu parametrów można oszacować wydajność protokołu, dzieląc czas potrzebny na przesłanie samej informacji przez całkowity czas transmisji, uwzględniający narzut protokołu. W przypadku protokołu AX.25, pracującego na łączu dwukierunkowym naprzemiennym, zależność tę można opisać następująco:

$$\eta = \frac{MaxFrame \cdot PaLen \cdot 8}{R_{wl}(AckTime + 2 \cdot TxDel) + \frac{64}{63} \cdot 160 + MaxFrame \cdot \frac{64}{63} \cdot (160 + 8 \cdot PaLen)}. \quad (7)$$

3.2.3. Wyniki obliczeń

Używając powyższych zależności, można oszacować efektywną przepustowość protokołu AX.25 dla różnych możliwych wartości *PacLen* (pojemność pola danych ramki informacyjnej) i *MaxFrame* (wielkość okna). Jako parametry *AckTime* oraz *TxDelay* przyjęto domniemane wartości używane w sterownikach TNC (ang. *Terminal Node Controller*) TNC3 firmy Symek [4]. Wynoszą one: *TxDelay* – 250 ms, natomiast *AckTime* – 280 ms dla prędkości łącza radiowego 9,6 kb/s i 2247 ms dla 1,2 kb/s. Wyniki obliczeń pokazano na rys. 4-5 dla prędkości łącza radiowego 1,2 kb/s i rys. 6-7 dla prędkości 9,6 kb/s.

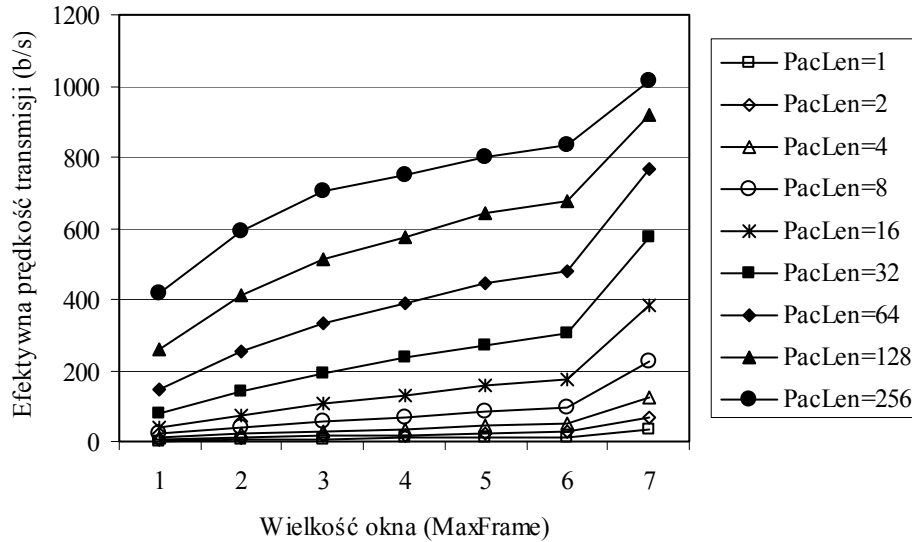
Warto zauważyć, że na wykresach widać dużą różnicę między wartościami uzyskanymi dla *MaxFrame* 6 i 7. Jest ona spowodowana właśnie czasem *AckTime*, który nie występuje przy *MaxFrame* równym 7, ponieważ protokół AX.25, podobnie jak i HDLC, nie dopuszcza większej liczby ramek informacyjnych w ciągu. Gdyby, przy mniejszej wielkości okna, nadawca mógł oznaczyć ostatnią wysyланą ramkę w ciągu (przez ustawienie bitu sterującego Poll/Final), różnica byłaby znacznie mniejsza, a wydajność protokołu – większa¹.

Używając przedstawionych zależności, można także oszacować maksymalną wydajność protokołu dla różnych prędkości łącza radiowego. Największa wydajność można zawsze uzyskać (teoretycznie) dla wartości parametrów *PacLen*=256 oraz *MaxFrame*=7. Jedynymi parametrami, których wartości będą się zmieniać, są zatem prędkość łącza radiowego (*R_{wl}*) oraz *TxDelay*. Zależność wydajności protokołu AX.25 od wartości *TxDelay* przy różnych prędkościach transmisji pokazano na rys. 8.

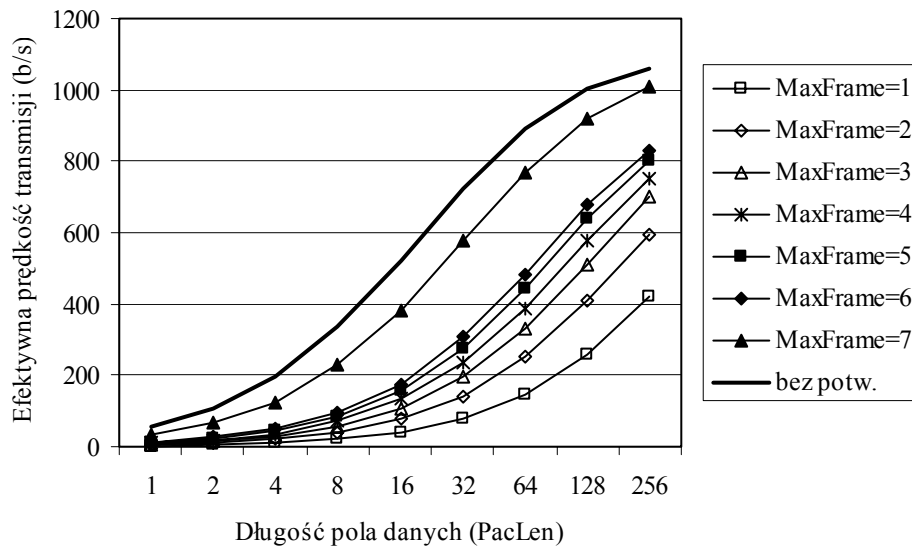
Na przedstawionym wykresie widać, że zwiększanie wartości *TxDelay* zmniejsza wydajność protokołu. Zależność ta jest jednak silnie uzależniona od prędkości transmisji. Dla wartości *TxDelay*=0, niezależnie od prędkości transmisji, protokół uzyskuje wydajność rzędu 90%. Dla niskich prędkości transmisji, nie przekraczających 9,6 kb/s, wydajność protokołu zmniejsza się niemal liniowo wraz ze wzrostem wartości parametru *TxDelay*. Niestety, dla większych prędkości transmisji, degradacja wydaj-

¹ „Umiejętność” oznaczenia ostatniej ramki w ciągu zależy od użytego oprogramowania, sterującego pracą kontrolera TNC. Na ogół oprogramowanie realizujące zbiór poleceń TAPR oznacza ostatnią ramkę, natomiast TF – nie.

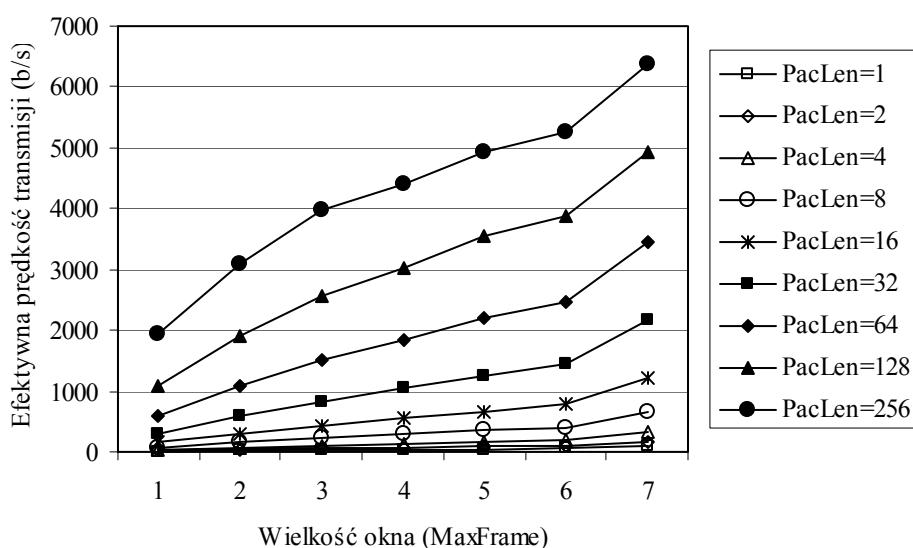
ności postępuje znacznie szybciej – przykładowo, dla prędkości 614,4 kb/s, już przy $TxDelay=5$ ms, wydajność protokołu spada poniżej 20%. Warto przy tym zauważyć, że tylko niektóre urządzenia pracujące w sieci Packet Radio mogą pracować przy tak małej wartości tego parametru – typowe wartości zależą od użytego nadajnika-odbiornika radiowego i wynoszą najczęściej 100-200 ms. Z drugiej strony, prędkości transmisji przewyższające kilkadziesiąt kb/s spotykane są rzadko – łącza naziemne pracują zazwyczaj z prędkością 9,6 kb/s, satelitarne zaś – 38,4 kb/s (ale tylko w kierunku „w dół”, od satelity do odbiornika naziemnego).



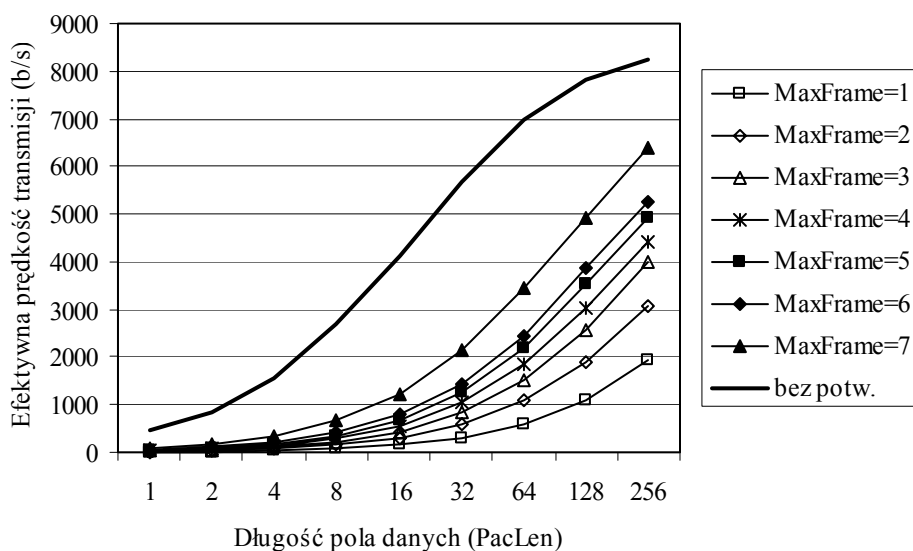
Rys. 4. Wpływ wielkości okna na efektywną prędkość transmisji dla łącza 1,2 kb/s



Rys. 5. Wpływ długości pola danych na efektywną prędkość transmisji dla łącza 1,2 kb/s

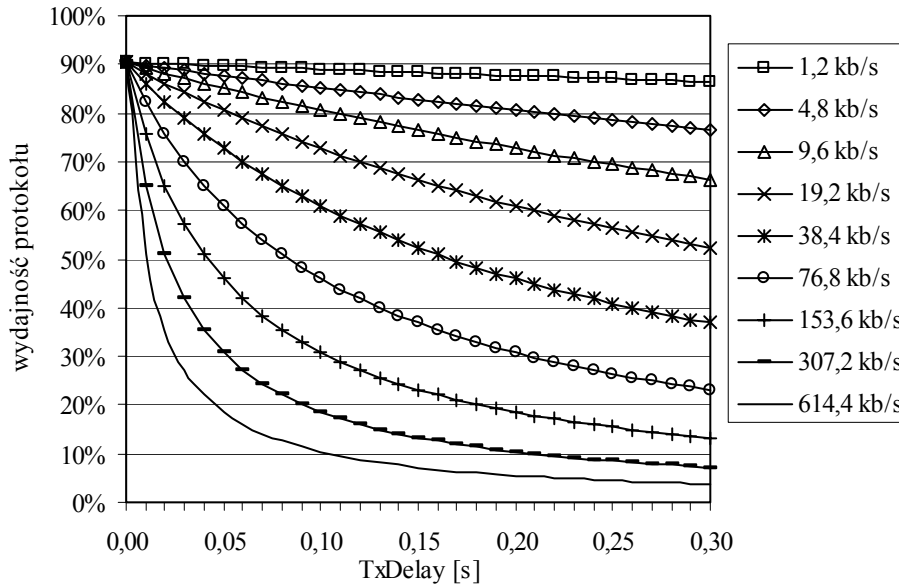


Rys. 6. Wpływ wielkości okna na efektywną prędkość transmisji dla łącza 9,6 kb/s



Rys. 7. Wpływ długości pola danych na efektywną prędkość transmisji dla łącza 9,6 kb/s

Warto zauważyć, że degradacja wydajności protokołu zmniejsza także efektywną prędkość transmisji. Nawet dla największych dopuszczalnych wartości parametrów ($PacLen=256$, $MaxFrame=7$), przy $TxDelay=250$ ms, efektywne prędkości transmisji wynoszą od 6,6 kb/s (na łączu radiowym 9,6 kb/s) do 27,3 kb/s (na łączu 614,4 kb/s). Widać zatem, iż – wbrew obiegowej opinii – nawet bardzo duża (jak na możliwości sieci Packet Radio) prędkość transmisji, za jaką można uważać 614,4 kb/s, zapewnia znacznie mniejszą przepustowość od łącza RS-232, używanego najczęściej w celu podłączenia sterownika TNC do komputera.



Rys. 8. Wpływ parametru $TxDelay$ na wydajność protokołu dla różnych prędkości transmisji

3.3. Łącze w pełni dwukierunkowe

Przedstawiona powyżej sytuacja zmienia się diametralnie, gdy łącze radiowe jest w pełni dwukierunkowe (ang. *full duplex*). Wprawdzie potwierdzana jest wówczas każda ramka informacyjna indywidualnie, jednak potwierdzenie przesyła się (w osobnym kanale częstotliwościowym) podczas transmisji kolejnej ramki informacyjnej. Z punktu widzenia czasu transmisji istotne jest wówczas tylko ostatnie potwierdzenie, ponieważ będzie ono przesłane już po wysłaniu ostatniej ramki informacyjnej. Wobec tego czas transmisji na łączu dwukierunkowym będzie zbliżony do czasu transmisji prowadzonej bez potwierdzenia (rys. 5 i 7), jednak przy zachowaniu wiarygodności przesyłu.

Całkowity czas transmisji łączem dwukierunkowym można wyrazić wzorem:

$$T_p = TxDel + \left\lceil \frac{FileSize}{PacLen} \right\rceil \cdot \left(\frac{64}{63} \cdot \frac{160 + 8 \cdot PacLen}{R_{wl}} \right) + \left(\frac{64}{63} \cdot \frac{160}{R_{wl}} \right), \quad (8)$$

w którym środkowy składnik określa czas transmisji wszystkich ramek danych, ostatni zaś – czas transmisji ostatniego potwierdzenia. Z kolei efektywność protokołu można określić jako:

$$\eta = \frac{8 \cdot FileSize}{R_{wl} \cdot TxDel + \frac{64}{63} \cdot \left\lceil \frac{FileSize}{PacLen} \right\rceil \cdot (160 + 8 \cdot PacLen) + \frac{64}{63} \cdot 160}. \quad (9)$$

Przy odpowiednio długiej transmisji danych pierwszy i ostatni składnik czasu transmisji T_p jest pomijalny. Wówczas efektywność protokołu jest niezależna od prędkości transmisji łącza radiowego i dla najdłuższych możliwych ramek ($PacLen=256$) wyno-

si około 91,3%. Umożliwia to efektywne wykorzystanie wszystkich, nawet najwyższych prędkości transmisji – przy łączu radiowym 614,4 kb/s otrzymuje się efektywną prędkość około 560 kb/s. W takiej sytuacji wykorzystanie pojemności sieci jest możliwe jedynie wówczas, gdy prędkość komunikacji z komputerem wzrośnie powyżej możliwości standardu RS-232, bądź też gdy większa liczba stacji będzie utrzymywać wysoką aktywność wymiany informacji. Z tego powodu niektóre kontrolery TNC wyposażono w możliwość podłączenia do sieci Ethernet (umożliwia to także integrację sieci Packet Radio i Internetu) lub z wykorzystaniem interfejsu USB. W obu przypadkach pozwala to na wystarczające zwiększenie prędkości transmisji.

3.4. Wpływ kontrolerów TNC na czas transmisji

Powyższe oszacowania dotyczyły samego protokołu AX.25. W przypadku stosowania kontrolerów TNC należy jeszcze uwzględnić czas transmisji między komputerem a dołączonym do niego kontrolerem, tym bardziej, że prędkość ta nie jest duża (zazwyczaj nie przekracza 115,2 kb/s, a często jest dużo niższa). Załóżmy jednak – jak ma to miejsce w większości przypadków – że prędkość ta jest wyższa od prędkości łącza radiowego (a tym bardziej od jego prędkości efektywnej). Można wówczas przyjąć, iż transmisja pomiędzy kontrolerami rozpocznie się po przesłaniu do kontrolera co najmniej *PacLen* znaków. Przy odpowiedniej różnicy efektywnej prędkości transmisji łącza przewodowego i radiowego, podczas transmisji pierwszej ramki danych, w kontrolerze zgromadzone zostaną kolejne znaki, których liczba może wystarczyć do sformowania następnej ramki zawierającej *PacLen* znaków w polu danych. W tej sytuacji można wykorzystać ustawioną wielkość okna (parametr *MaxFrame*). Z kolei po stronie odbiorczej po zakończeniu transmisji między kontrolerami pozostanie jeszcze co najmniej *PacLen* znaków do wysłania łączem przewodowym.

Całkowity czas transmisji między komputerami będzie sumą następujących elementów:

- czasu transmisji łączem szeregowym co najmniej *PacLen* znaków z komputera nadającego do kontrolera TNC,
- czasu transmisji informacji łączem radiowym, zgodnie z zasadami protokołu AX.25 i ustalonymi parametrami transmisji, wyliczony według (6) lub (8),
- czasu transmisji łączem szeregowym co najmniej *PacLen* znaków z kontrolera TNC do komputera odbierającego.

Całkowity czas transmisji można zatem wyrazić wzorem:

$$T_s = \left\lceil \frac{FileSize}{PacLen \cdot MaxFrame} \right\rceil \cdot \left(AckTime + 2 \cdot TxDel + (1 + MaxFrame) \cdot \frac{64}{63} \cdot \frac{160}{R_{wl}} \right) + \left\lceil \frac{FileSize}{PacLen} \right\rceil \cdot \left(\frac{64}{63} \cdot \frac{8 \cdot PacLen}{R_w} \right) + 2 \cdot \frac{10 \cdot PacLen}{R_w}. \quad (10)$$

Ostatni składnik równania określa czasy transmisji danych między komputerami a kontrolerami TNC. Przyjęto, że na każdy znak przesyłany łączem szeregowym potrzeba 10 bitów, a prędkość transmisji wynosi R_w b/s. W zależności od szczegółów implementacji oprogramowania kontrolera TNC, jak np. czas przetwarzania danych,

ograniczenia pojemności buforów czy też możliwości wymuszania natychmiastowych potwierdzeń, sumaryczny czas transmisji może ulec dodatkowemu wydłużeniu.

4. Podsumowanie

W niniejszym rozdziale pokrótce opisano protokół AX.25 oraz zaprezentowano metodę szacowania jego wydajności przy różnych konfiguracjach sieci. Osiągnięte rezultaty są zbieżne z wynikami uzyskanymi w praktyce podczas testów kontrolerów TNC [5, 6, 7], co może świadczyć o wystarczającej dokładności dokonanego oszacowania. Może ono służyć zatem jako model opisujący zachowanie sieci, który pokazuje teoretyczne możliwości protokołu AX.25. W tej sytuacji uzyskane wyniki można zastosować jako „punkt odniesienia” dla rezultatów uzyskanych na drodze eksperymentów wykonanych w rzeczywistej, doświadczalnej sieci Packet Radio [8].

Literatura

1. Dąbrowski A.: Amatorska komunikacja cyfrowa. Warszawa, PWN, 1994.
2. Beech W. A., Nielsen D. E., Taylor J.: AX.25 Link Access Protocol for Amateur Packet Radio. Tucson Amateur Packet Radio Corporation, 1997.
3. Wolisz A.: Lokalne sieci komputerowe. Tom 1: Sprzęt sieciowy. Warszawa, WNT, 1993.
4. Kumm U.: Software-User-Manual for the 16 bit High-Speed Packet-Radio-Controller TNC3S, TNC31S. SYMEK GmbH, Stuttgart, 1999.
5. Zieliński B.: Using TNC controller as a model of protocol converter. W: Pułka A., Hryniewicz E., Kłosowski P. (eds.): Proceedings of IFAC Workshop on Programmable Devices and Systems PDS2004, Kraków 2004, pp. 269-273.
6. Zieliński B., Tokarz K.: Wpływ mocy obliczeniowej konwertera protokołów na wydajność transmisji. *Studia Informatica*, Vol. 21, Number 1 (39), pp. 171-185.
7. Zieliński B.: Influence of protocol converter processing power upon network efficiency. W: Bradac Z., Zezulka F., Polansky M., Jirsik V. (eds.): Proceedings of IFAC Workshop on Programmable Devices and Embedded Systems PDeS 2006, Brno 2006, pp. 38-43.
8. Zieliński B.: Porównanie różnych wersji kontrolerów TNC. W: Gaj P., Pochopień B., Kozielski S. (eds.): Współczesne aspekty sieci komputerowych. Tom 1. WKŁ, Warszawa 2008, rozdz. 13, pp. 137-146.