EFFECTIVE TRANSMISSION SPEED IN AX.25 PROTOCOL

Bartłomiej M. Zieliński, *Non-Member*

Abstract: **The paper considers factors that have influence upon effective transmission speed using AX.25 protocol. Basing on brief description of the protocol and transmission hardware, analytical equations are derived that make it possible to calculate effective transmission speed in few cases. The paper shows results of these calculations and compares them with those obtained in a practical tests. The paper ends with brief discussion on achieved results.**

Index Terms**: Packet Radio network, AX.25 protocol, effective transmission speed**

I. INTRODUCTION

AX.25 protocol is used in amateur Packet Radio network on data link layer. It is a modified variant of widely known HDLC protocol. As such, it contains not only data link layer mechanisms, but also some elements typical for network layer (static routing) and transport layer (flow control).

Packet Radio network is an example of simple wireless wide area network. It was created in early 1980's, when Internet and cellular telephony were not widely available yet. Packet Radio, as a work of radio amateurs, was never very popular and there are only very few papers on this subject (e.g., [3]). Currently, Packet Network is often used to transmit telemetry data according to APRS protocol [4].

This paper extends the results presented in [5], [6].

II. AX.25 PROTOCOL AND TNC CONTROLLERS

AX.25 protocol [1], used as a data link layer in Packet Radio network, belongs to the HDLC protocol family and mostly resembles LAP-B. The modifications result from radio amateur operation and comprise mostly address field that is capable of carrying up to 10 station addresses of 7 bytes each [1]. Such a large address part was necessary in earlier implementations of Packet Radio networks, where no network layer was present, hence routing had to be done statically on data link layer level. Nowadays, when several network layers may operate over AX.25 protocol, number of station addresses may be reduced to 2.

A) Frame format description

General frame format in AX.25 protocol is presented on fig. 1.

Similarly to HDLC, every frame begins and ends with a flag of constant '01111110' value.

Address field contains 2 to 10 station addresses, i.e., sender and receiver and up to eight optional intermediate stations addresses. Every address contains up to 7 bytes, which results from using amateur "call signs" as station addresses. Thus, the shortest address field contains 14 bytes, the longest one – 70 bytes.

Control field specifies frame type and, for some types, frame sequence numbers. There are I (information), C (control) and U (unnumbered) frames. I-type frames are generally used to carry user data after a logical connection is established. All such frames must be positively acknowledged; otherwise, limited number of retransmission attempts shall occur. U-type frames are used for connection management. Some of them may also carry user data, typically before a logical connection is established; such transmissions are not acknowledged, but allow for broadcast transmissions. Control frames are used mostly for acknowledgments. There are few types of acknowledgement possible:

- RR (Receiver Ready) – positive,
- RNR (Receiver Not Ready) – positive, but allows receiver to pause transmission for some time,
- REJ (Reject) – transmission error,
- FRMR (Frame Reject) – serious error, unexpected frame type.

RR and RNR frames may acknowledge several I-type frames (up to maximum window size set), thus reducing protocol overhead.

PID (Protocol ID) field specifies if there is any network layer above AX.25 and what type it is. One of such layers may be TCP/IP protocol stack.

Data field is present only in I and U-type frames and may contain up to 256 bytes. Its maximum size may be further limited by a N_1 parameter, depending on actual link quality and other properties.

CRC field is calculated according to CRC-CCITT algorithm. Depending on whether calculated and received CRCs are equal, a positive or negative acknowledge is sent. Similarly to HDLC, up to 7 frames may be transmitted before a positive acknowledge is sent; this number (window size) may be further limited by a k parameter, depending on link quality. Actual number of frames depends on data format and transmission equipment possibilities.

In order to preserve protocol transparency that might be violated by using special bit sequences for

Flag (BOF)	Addresses	Control	PID	Data	CRC	Flag (EOF)
1 B	14 to 70 B	1 B	1 B	Up to 256 B	2 B	1 B

Fig. 1 – General frame format in AX.25 protocol

preamble and postamble, a technique call "bit stuffing" is used that comprises entire frame except preamble and postamble. After every group of five bits equal to '1', one '0' bit is automatically inserted into bit stream. In the receiver this operation is reverted.

B) Protocol parameters

AX.25 protocol and TNC controller behaviour may be adjusted by large number of parameters (depending on TNC software). From the point of view of protocol efficiency, the most important are the following:

- k – maximum number of frames sent consecutively before waiting for an acknowledge (window size); not larger than 127 in protocol version 2.2 (still rarely implemented) and 7 in older versions;
- N_1 – maximum capacity of data field in an information or unnumbered frame; not larger than 256;
- T_2 – time that elapses between the end of latest I frame and acknowledge; depending on software and parameters, this delay may or may not occur during transmission and may be set manually or automatically;
- T_{103} – time for stabilisation of transmitter parameters after transmission start and for carrier detection in receiver; depends on radio transceiver capabilities and varies from few tens to few hundreds milliseconds;
- T_{102} – slot duration for carrier sense-based collision avoidance; typically about 50 to 300 ms;
- p – persistence parameter of carrier sense-based collision avoidance (not defined by protocol description but implemented in most of the software); typically equal to 64 which means transmission probability of 25%.

C) Frame transmission process

Before the transmission starts, the station must check whether the channel is free. This is done by p-persistent carrier sense mechanism. For any given value of p and T_{102}, one can calculate average medium access delay as

$$T_{CS} = \frac{256T_{102}}{2(p+1)}. \qquad (1)$$

Afterwards, transmitter is turned on. TNC waits for T_{103} to ensure that transmitter works stable and then sends information (I) frame. If window size (k) is larger than one and there are enough frames ready for transmission, several I frames may be sent consecutively. When the transmission stops, the receiver waits for T_2 to ensure there are no more I frames, turns on the receiver, waits for T_{103} and sends the RR frame (acknowledge). Transmission process for window size of 1 and 4 is shown on fig. 2 and 3, respectively.

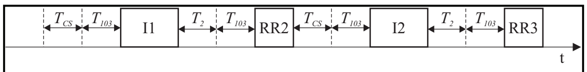

Fig. 3 – Frame exchange when k=1

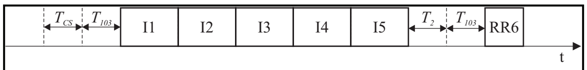

Fig. 2 – Frame exchange when k=4

Some comment is necessary on T_2 delay. Depending on the TNC software and AX.25 protocol implementation details, the sender may request immediate acknowledge by setting a special control bit (Poll/Final) in the control field of a frame. Only the latest I frame within a window should be marked this way. In such a case, the receiver knows there will be no more frames and responds with RR immediately, not wasting T_2 for unnecessary waiting for more frames. What seems interesting, such behaviour is not required by AX.25 definition and Poll/Final bit usage in this context depends on TNC software.

D) TNC controllers

TNC controller (*Terminal Node Controller*) is a microprocessor-based circuit that provides for connection of DTE device (e.g., personal computer) to Packet Radio network [1].

TNC controller consists of a two parts. In the digital one, data format processing is run according to the requirements of Packet Radio network and operating rules of AX.25 protocol. In turn, analogue part plays a role of a modem and provides for a control over radio transceiver directly from TNC. Block diagram of the controller is shown of fig. . It might suggest that every element shown on this diagram must be a separate circuit. In fact, using most up-to-date technologies and modern single-chip microcontrollers, entire digital part may be implemented in a single IC. Some TNC controllers, however, require attachment of external program and data memories. It depends mostly on the architecture and capabilities of the microcontroller. On the other hand, even in the oldest controllers, serial port and HDLC controller were placed in a single IC (in most cases, Z80-SIO or 8530).

Currently available TNC controllers may be divided into several groups. The most often met are circuits compatible with TNC2 standard, built using Zilog Z80 microprocessor. Much more modern are TNC3 circuits, in which single-chip version of a popular Motorola 68000 microprocessor was used. There are also intermediate circuits, containing Zilog Z180 microprocessor or – much more often – Motorola 68HC11 or 68HC12 single-chip microcontrollers. The most modern circuits are built using 32-bit single-chip microcontrollers with ARM7 processor core, such as LPC2000.

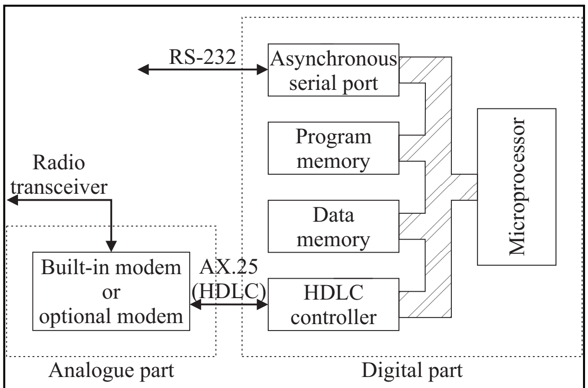

Fig. 4 – Structure of TNC controller hardware

Software of TNC controller is responsible for proper realisation of AX.25 protocol that is used as a data link layer in Packet Radio network. Other functions comprise communication with DTE (and operator) and controller parameters configuration. It is worth notice that while realisation of AX.25 protocol should be strictly standardised (some differences, however, can be observed), remaining functions may be implemented more freely. Hence there are several methods of computer-to-controller communication. Some of them are optimised for human-to-human communications (TAPR and TF command sets) and transmission between devices (KISS and HOST modes and Hayes AT command set in TNC3). Availability of operating modes depends on hardware layer of the controller – the widest range of software types and versions are for Z80-based controllers compatible with TNC2 standard. It is also interesting that European controllers are usually equipped with TF software, while American ones – with TAPR.

The type of software used may have some influence on details of realisation of specific functions, especially those related to AX.25 protocol. We may therefore expect that controller efficiency depends not only on processing power resulting from microprocessor type and speed, but also on type and version of software used.

III. EFFECTIVE TRANSMISSION SPEED

In this section, we show analytical equations that allow estimate transmission time in Packet Radio network using AX.25 protocol. Both half-duplex and full-duplex radio links are considered. In addition, we consider transmission using TNC controllers which are often used as transmission hardware in Packet Radio network.

For the purpose of analysis, let's assume that:

– Transmission proceeds between two stations only,
– There are no collisions or transmission errors,
– Frame processing time is negligible,
– entire user data is L bytes long,
– radio link operates at R_{wl} bps,
– wired link operates at R_w bps,
– RR acknowledgement frame consists of 20 bytes,
– Information frame contains $20{+}N_1$ bytes.

We take into account the following protocol parameters:

– N_1 – maximum capacity of a data field,
– k – maximum window size,
– T_{103} – transmitter startup timer,
– T_2 – response delay timer.

A) AX.25 protocol – half-duplex link

On half-duplex radio link, transmission consists of several cycles. Every cycle consists of up to k information frames and the RR acknowledge. Number of cycles depends on actual transmission parameters, such as k, $N1$ and L.

It may be proved that transmission cycle, that consists of k information frames and the RR acknowledge, lasts

$$T_p = T_2 + 2T_{103} + (1+k)\frac{64}{63}\frac{160}{R_{wl}} + k\left(\frac{64}{63}\frac{8N_1}{R_{wl}}\right). \tag{2}$$

The first component of (1) is protocol overhead, while the second one determines transmission time of user data.

If entire data is L bytes long, it transmission time may be expressed as

$$T_f = \left[\frac{L}{N_1}\right]\left(\frac{64}{63}\frac{8N_1}{R_{wl}}\right) + \left[\frac{L}{N_1 k}\right]\left(T_2 + 2T_{103} + (1+k)\frac{64}{63}\frac{160}{R_{wl}}\right). \tag{3}$$

B) AX.25 protocol – full duplex link

On full-duplex radio link, each information frame is acknowledged individually on a separate radio channel. Thus, acknowledgements are not significant for transmission time. User data transmission time may be therefore expressed as

$$T_f = T_{103} + \left[\frac{L}{N_1}\right]\left(\frac{64}{63}\frac{160+8N_1}{R_{wl}}\right) + \frac{64}{63}\frac{160}{R_{wl}}. \tag{4}$$

C) TNC controllers

If TNC controllers are used, T_f specifies only transmission time between them. However, user data must also be transmitted between TNC's and attached devices. We may assume that at least N_1 bytes must be transmitted before sender-side TNC starts radio transmission. Similarly, the recipient-side TNC sends at least N_1 after radio transmission stops. Thus,

$$T_t = 2 \cdot \frac{10 \cdot N_1}{R_w} + T_f. \tag{5}$$

T_f must be calculated using (2) or (3), depending on the radio link type – half-duplex or full-duplex, respectively.

D) Result examples

Using above equations, one can calculate effective throughput of AX.25 protocol for various possible values of N_1 and k. As values of T_{103} and T_2, default parameters used by TNC3 controller were accepted (T_{103} – 250 ms, T_2 – 280 ms for 9600 bps and 2247 ms for 1200 bps). Calculation results are presented on figures 4–5 for 1200 bps radio link and figures 6–7 for 9600 bps radio link.

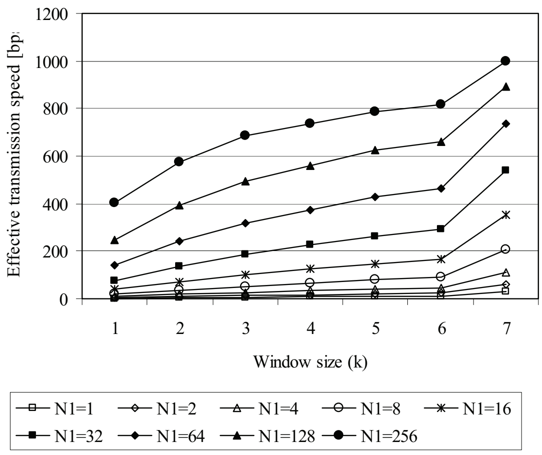

Fig. 5 – Effective throughput as a function of k for various N_1 (1200 bps)

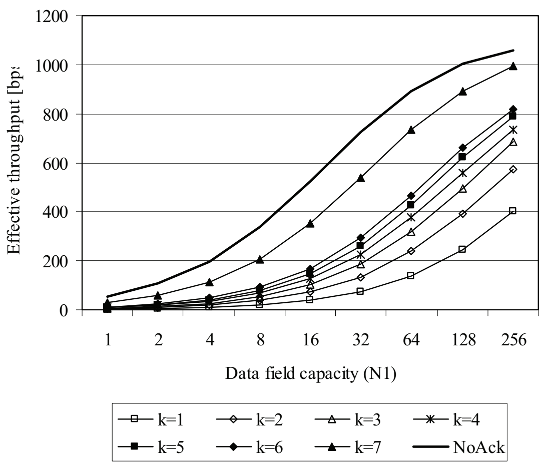

Fig. 6 – Effective throughput as a function of N_1 for various k (1200 bps)

As expected, it was found that the network efficiency was higher for longer data frames as well as for window size. For wireless link operating at 1200 bps the maximum effective transmission speed (observed by the user) equals to about 1000 bps. Without acknowledgements, similarly to full-duplex link, it is a little higher and reaches about 1050 bps. For wireless link operating at 9600 bps, the effective speeds are about 6000 and 8200 bps, respectively. In this case, the difference is much bigger, which may result from the T_{103} parameter, the influence of which rises with increasing radio link bit rate.

It is worth notice that the results for both rates show big difference between k values of 6 and 7. This is caused by T_2, which does not count for k=7 as there are never more than 7 consecutive data frames possible. If the sender could mark the last data frame with Poll/Final bit, this difference would be much smaller.

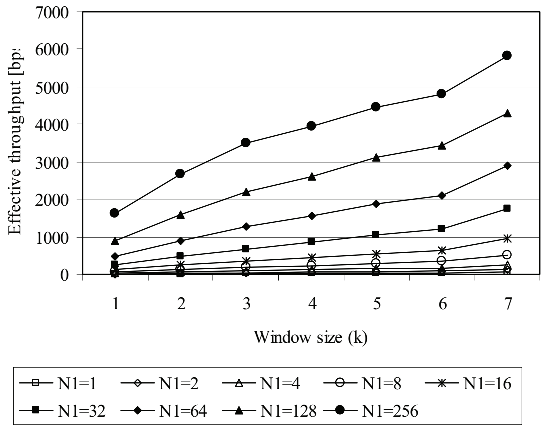

Fig. 7 – Effective throughput as a function of k for various N_1 (9600 bps)

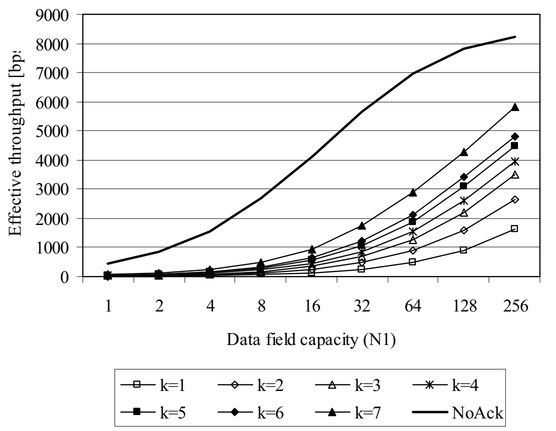

Fig. 8 – Effective throughput as a function of N_1 for various k (9600 bps)

E) Experimental results

The analytical results shown above are close to the values obtained in an experimental network. It must be noticed however that the analytical results correspond to perfect conditions and do not consider some factors that might influence the efficient throughput of the network. Such factors are, among others, protocol implementation details and processing power of network hardware. Microprocessor types used in the TNC controllers are collected in table 1.

Table 1. Microprocessors in TNC controllers

Controller	Processor	f_{clk} [MHz]
TNC2	Z80	2.4576
TNC2D	Z80	4.9152
Spirit-2	Z80	19.6608
KPC-9612+	68HC11	16.0000
TNC3	68302	14.7456
TNC7	LPC2106	58.9824
DLC7	S3C4530	49.1520

Tests of TNC controllers efficiency were conducted in an experimental network, containing one or two PC-class computers and two controllers. A single PC computer is sufficient if it is equipped with two RS-232 ports or USB, depending on controllers used in a given test. The controllers were connected with cables. This exotic – as for circuits designed for wireless communication – configuration was chosen in order to avoid negative influence of radio interference over transmission quality. Besides, in such a network it was possible to set any parameter values freely, which allows for testing cases not very common in practice. As the radio transceiver is always under full control of TNC, lack of transceiver does not influence transmission time. Network configuration is shown on fig. 3.

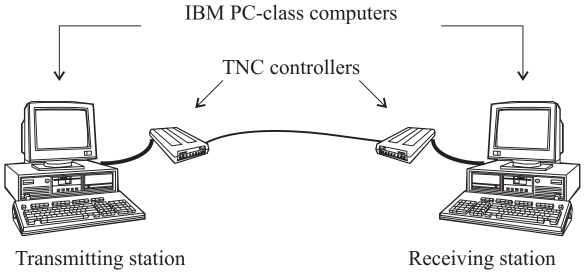

Fig. 9 – Experimental Packet Radio network

The tests were conducted, transmitting a file of 8 or 16 KB (depending on transmission rate) for various values of window size and data field capacity of AX.25 protocol information frame. The controllers operated in half-duplex Asynchronous Balanced Mode, as is typical for Packet Radio communication.

Measurements results of effective transmission speed for few selected TNC controllers, operating at various window sizes and maximum length data frames (256 B of data) are shown on fig. 10. Transmission rate was equal to 19.2 kbps on serial port and 1.2 kbps on radio link. For comparison, the graph contains also a curve determining theoretical capabilities of AX.25 protocol. On the graph we can see that the results do not differ very much. Some controllers (e.g., TNC2, TNC2D) can not make use of window size 4 and above – increasing this parameter does not practically increase transmission speed. KPC-9612 behaves similarly. Faster TNC3 and TNC7 controllers, unexpectedly, behave worse than the others for window sizes less than 7. A more detailed analysis conducted in monitoring mode showed that these controllers did not request immediate acknowledgement by setting Poll/Final bit in AX.25 protocol control field. Thus, the recipient waits for *T2* time for possible consecutive frames and sends the acknowledgement only afterwards. However, when window size equals to 7, TNC3 and TNC7 achieve highest throughput than other controllers, close to theoretical values. It is possible, because, for maximum window size allowed by protocol definition, recipient does not count *T2* time.

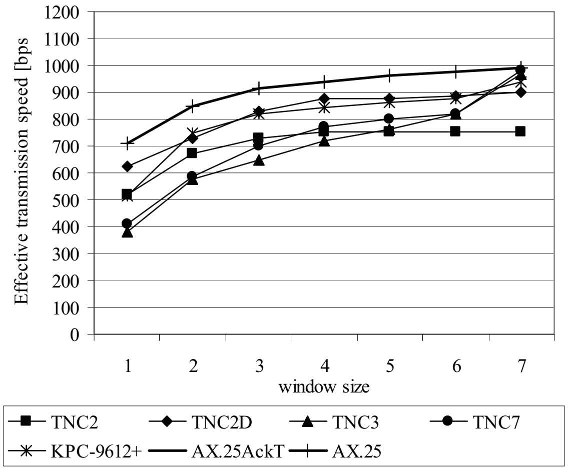

Fig. 10 – Effective transmission speed on 1200 bps radio link

Results of similar measurements, conducted at radio link transmission rate of 9.6 kbps, are presented on fig. 11. In this case, difference between various TNC controllers is much more visible. Depending on controller type, maximum effective speed varies from about 1.5 kbps (TNC2D) to almost 5 kbps (TNC7). For comparison, the graph presents two curves showing theoretical capabilities of AX.25 protocol. One of them (AX.25 imm) corresponds to immediate acknowledge case, the second one (AX.25 T2) – with respect to *T2* time.

Presented results show that effective transmission speed achieved in a given configuration depends on not only hardware – particularly microprocessor type and its clock frequency – but also properties of software that controls TNC controller operation. The following factors, depending exclusively on software, may have influence over circuit performance:

- full utilisation of window size for every data field capacity of a frame,
- sufficiently high AX.25 protocol frames processing speed,
- immediate (without delays) proper information

frame reception acknowledgement generation,

- immediate acknowledgement request by setting of Poll/Final bit in the latest information frame within a window.

Inability of full utilisation of window size is especially annoying in Z80-bases TNC controllers, practically regardless of its clock frequency and memory capacity. However, type and version of software used in TNC has some influence upon its performance. For example, versions supporting TAPR command set utilise window size rarely – the controller can not transmit more than 5 maximum-length frames consecutively. A little better is TF software, which, especially in most up-to-date 2.7 version, can send up to 7 maximum-length frames consecutively. It seems however, that such capability is achieved at a cost of longer frames preparation for transmission.

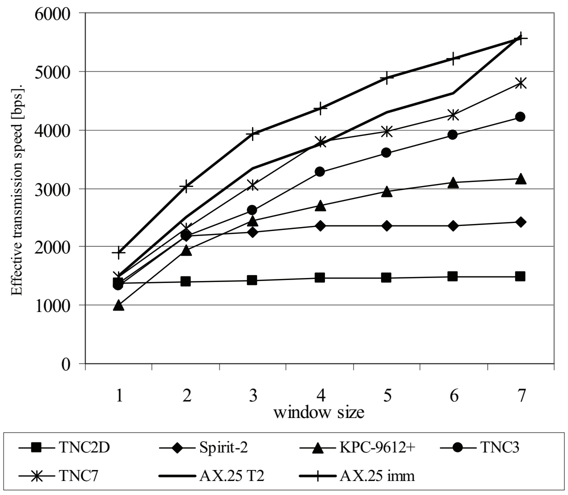

Fig. 11 – Effective transmission speed on 9600 bps radio link

Controllers, supporting TAPR command set, but based on other microprocessor types, can utilise window size much better, even at higher transmission rates. However, there is no other software for these controllers, it is thus hard to say if this capability results from higher processing power of a microprocessor, or better software optimisation.

Additional factor that influences effective transmission speed is the way the recipient treats window size less than 7. If the sender does not mark the latest frame within a window with Poll/Final bit, TF software sends the acknowledgement only after *T2* time elapses, while TAPR – immediately. In some versions of TF software *T2* time may be set, in others – e.g., TNC3 – it is calculated automatically and cannot be changed. This parameter can also be set in some versions of TAPR software.

Some software versions, at the beginning of transmission, initially limit window size, and later gradually increase it up to maximum value set. Such behaviour may be reasonable, because it allows recognise capabilities of a receiving station. However, when the transmitted information is relatively short, the transmission efficiency decreases.

IV. CONCLUSIONS

In the paper we briefly described AX.25 protocol and TNC controllers. We showed analytical equations that allow estimate efficient transmission speed of AX.25 protocol in various conditions. Results of calculations based on these equations are then compared to those obtained in an experimental network. They show that not only hardware properties have influence upon network efficiency, but also – in at least equal degree – properties of software that drives TNC controller. We pointed implementation details of AX.25 protocol, used in Packet Radio network, that have significant influence upon effective transmission speed. These observations may show directions of existing software modifications. It is essentially important for TF software, a source code of which is available in Internet.

A more detailed comparison of properties of TNC control software may be obtained, when there is a hardware platform for which there are many software types and versions available. Currently, only Z80-based TNC controller family forms such a platform.

REFERENCES

[1]. Beech W. A., Nielsen D. E., Taylor, J. AX.25 Link Access Protocol for Amateur Radio. Tucson Amateur Packet Radio Corporation, Tucson 1997.

[2]. Dąbrowski A. Digital amateur communications. PWN, Warszawa, 1994 (in Polish).

[3]. Karn P. R., Price H. E., Diersing R. J., “Packet Radio in the Amateur Service”, IEEE JSAC, Vol. 3, No. 3, May 1985, pp. 431-439.

[4]. Wade I. (ed.). Automatic Position Reporting System. APRS Protocol Reference. Protocol Version 1.0. Tucson Amateur Packet Radio Corporation, Tucson, 2000.

[5]. Zieliński B. Using TNC controller as a model of protocol converter. In Pułka A., Hrynkiewicz E., Kłosowski P. (ed.), Proceedings of “IFAC Workshop on Programmable Devices and Systems PDS2004”, Kraków, Poland, 17-19 November 2004, pp. 269-273.

[6]. Zieliński B. Influence of protocol converter processing power upon network efficiency. In Bradac, Z., Zezulka, F., Polansky, M., and Jirsik, V. (ed.), Proceedings of “IFAC Workshop on Programmable Devices and Embedded Systems PDeS 2006”, Brno, Czech Republic, 11-14 February 2006, pp. 38-43.

Bartłomiej Zieliński works in the Silesian University of Technology, Institute of Computer Science. He achieved MSc degree in 1993 and PhD in 1997. His professional interests cover data link and medium access layer protocols of wireless networks as well as microprocessor and embedded systems.