

# The Importance of Sliding Window Protocol Generalisation in a Communication Protocols Course

## Author:

Drago Hercog, University of Ljubljana / Faculty of Electrical Engineering, Ljubljana, Slovenia,  
Drago.Hercog@fe.uni-lj.si

**Abstract** — All modern communication systems rely on the correct operation of communication protocols that are running in different layers of protocol stacks. Hence, a deep knowledge and understanding of them must be possessed by all communications engineers. Therefore, a basic communication protocols course must be an essential constituent of any higher education communications engineering programme. Automatic repeat request (or ARQ for short) protocols that provide for reliable data transfer over noisy and lossy channels and are usually used in data-link and transport layers of a protocol stack often form the core of such courses. One of the reasons for their high pedagogic importance within the frame of a protocols course is that they are typical but not trivial distributed algorithms and therefore not very easy to grasp; the other reason is that they must be both logically correct and efficient. Pedagogically, the general notions of correctness and efficiency can easily be introduced and emphasised through the correctness and efficiency of ARQ protocols. Automatic repeat request protocols are usually implemented in the form of sliding window protocols. Traditionally, three basic types of sliding window protocols are distinguished, namely, stop-and-wait, go-back-N, and selective-repeat protocols. Although all the three types were often claimed to be members of a common sliding window protocols family, this was not quite clear and evident until recently, which was quite confusing for both students and teachers. However, several years ago, the generalised sliding window protocol was defined and formally specified so that the three above mentioned basic types can easily be shown as its special cases. The protocol behaviour depends on two essential parameters, namely transmit window width and receive window width which are directly related to the amount of memory used by transmitter and receiver, respectively. In the generalised protocol, these parameters can have arbitrary values, while in the three special cases their values are differently restricted. When the three basic types of sliding window protocols are explained as special cases of the generalised protocol, the relation of their behaviours to memory usage at both transmitting and receiving sides is much easier to be seen. A simulator of the generalised sliding window protocol was implemented and is used for both pedagogic and research purposes, where any of the three basic protocol types can be chosen to be simulated simply by choosing appropriate values of transmit and receive window widths. The generalised protocol was shown to be potentially equally efficient as the basic types while using less resources. From the pedagogic viewpoint, it is extremely important to be aware that a systematic approach to this complex topic can be both much easier and more understandable for students.

**Index Terms** — Communication Protocol, Communication Protocol Course, Generalisation, Sliding Window Protocols.

## INTRODUCTION

All modern digital communication systems are built according to the layered architecture referred to as the protocol stack, with a layer-specific protocol running in each individual protocol stack layer [1]. A layer protocol thus implements the functionality of the layer. Many different protocols exist, implementing different functionalities. A communication protocol actually is a distributed algorithm; it is typically quite complex and hence difficult to be understood and even more to be correctly and efficiently designed, especially by a student.

A communication protocols course is therefore an essential constituent of a modern telecommunications engineering higher-education programme. Within such a programme, several aspects of communication protocols must be emphasised:

- distributed algorithmic nature of protocols,
- systematics of protocol presentation,
- specification of protocols,
- logical correctness of protocols,
- protocol efficiency,
- verification of protocol correctness, and
- verification of protocol efficiency.

An extremely important functionality of some communication protocols is to detect and correct data errors and losses that normally occur in noisy communication channels. A class of protocols that can do this job is represented by so-called Automatic Repeat reQuest (or ARQ for short) protocols that are able to detect transmission errors and losses and retransmit corrupted or lost packets. ARQ protocols are normally implemented as sliding window protocols. Sliding window protocols are real-life protocols with their complexities spanning from simple to quite complex. All the aspects mentioned in the previous paragraph are very relevant for this family of protocols. For all these reasons sliding window protocols usually form the core of a communication protocols course.

Traditionally, only three special types of sliding window protocols have been presented to telecommunications students. However, a generalised protocol was recently defined, which allows sliding window protocols to be presented more systematically. In this paper, the role of the generalised protocol as well as its special cases within a communication protocols course will be discussed.

## SLIDING WINDOW PROTOCOLS

A sliding window protocol is an automatic repeat request (ARQ) protocol to provide for reliable data transfer over a lossy channel. Basically, as with any ARQ protocol, the receiver sends an acknowledgment for any correctly received packet. If the acknowledgment is not received by the transmitter within specified time period (because either the data packet or acknowledgment itself was lost), the transmitter's timer expires and the transmitter resends the presumably lost data message. The name "sliding window" refers to the management of the memory buffers that are used by the transmitter to store the outstanding (i. e., already sent but not yet acknowledged) packets, and by the receiver to store the packets that have already been received but are not yet ready to be passed to the user. The ranges of packets that can be stored into these buffers are called the transmit and the receive window, respectively, and the maximum numbers of packets that can be stored into the buffers are referred to as the transmit and the receive window width, respectively. In the course of the protocol operation, both windows slide along the packet sequence number scale.

Traditionally, three basic types of the sliding window protocol have been used and also taught in the communication protocols courses. These are the stop-and-wait, go-back-N, and selective-repeat protocols [2].

The stop-and-wait protocol where both window widths equal 1, allows the transmitter to have only one outstanding packet at a time; this protocol is very simple, requires a minimum amount of memory, but is unfortunately very inefficient because the transmitter may have to wait for acknowledgments most of the time.

The go-back-N protocol has the transmit window width greater than 1, which allows the transmitter to send several (up to the number equal to the transmit window width) packets without having to wait for acknowledgments; however, if the transmit window width is full the transmitter must wait for a new acknowledgment. Because the receive window width is 1 the receiver has no room to arrange packets that might come out of order and can accept only the packets that are received in order; therefore, if a packet is lost the transmitter must resend that packet and all those that were already transmitted after the presumably lost packet.

The selective repeat protocol also has the transmit window width greater than 1 and behaves identically to the go-back-N protocol in the absence of loss. However, if a packet is detected to have been lost the transmitter retransmits only that particular packet; the retransmitted packet thus is received out of order by the receiver, but the receiver has sufficient room to reorder the received packets, as its receive window width is equal to the transmit window width.

Both go-back-N and selective-repeat protocols are much more efficient than the stop-and-wait protocol if the transmit window width is large enough. However, the efficiency of the selective-repeat protocol is higher than the efficiency of the go-back-N protocol in most cases.

The three above mentioned protocols are treated as the members of the sliding window protocols family [2], differing in their respective transmit and receive window widths, as shown in Table 1. It is clear that the stop-and-wait protocol is a special case of both go-back-N and selective-repeat protocols and also that the behaviour of go-back-N and selective-repeat protocols is identical in the absence of loss. However, the respective behaviours of go-back-N and selective-repeat protocols in the presence of loss are quite different.

Protocol	Transmit Window Width	Receive Window Width
Stop-and-Wait	1	1
Go-Back-N	$W$	1
Selective-Repeat	$W$	$W$

TABLE 1  
WINDOW WIDTHS OF SLIDING WINDOW PROTOCOLS

Due to the above mentioned difference between go-back-N and selective-repeat protocols it is not possible to specify both of them with a single specification. Hence it is not easy to systematically present the above three protocols as the members of a single protocol family to the students. This pedagogic problem stimulated the author to search for a possibility to present the sliding window protocols more systematically. The result of this research led to the definition of the generalised sliding window protocol [3].

## GENERALISATION OF SLIDING WINDOW PROTOCOL

The generalised sliding window protocol was first described in [3] and formally specified in [4] using the Specification and Description Language (SDL) [5]. This protocol uses a combination of go-back-N and selective-repeat retransmission strategies. Stop-and-wait, go-back-N and selective-repeat protocols are special cases of the generalised protocol. Any combination of transmit and receive window widths can be used with the generalised protocol; however, the case with both window widths equal to 1 yields the stop-and-wait protocol, the case with the receive window width equal to 1 yields the go-back-N protocol, and the case with both window widths being equal yields the selective-repeat protocol. Although the case where the receive window width is larger than the transmit window width is logically correct, it is senseless as it does not provide for any higher efficiency than the selective-repeat case while using more resources.

Based on the formal specification of the generalised protocol and the above mentioned relations on transmit and receive window widths, stop-and-wait, go-back-N and selective-repeat protocols can be formally shown to be special cases of the generalised protocol. Moreover, a performance simulator was derived out of the formal generalised protocol specification; it can be used to simulate the stop-and-wait, go-back-N, selective-repeat, or generalised protocol simply by choosing appropriate transmit and receive window widths.

The sliding window protocol is based on a few basic communication mechanisms (namely, acknowledgments of correctly received data messages, sequence numbering of data packets and acknowledgments, and the use of a timer) which are all necessary for logical correctness of the protocol. In practice, some additional mechanisms are usually used only to improve the protocol efficiency (most usually, negative acknowledgments are used). However, the basic protocol represents the theoretical basis for all practically used protocols, which can be easily derived from it. Therefore, the basic protocol is normally taught in details to students within the frame of higher-education communications curricula.

With the introduction of the generalised sliding window protocol, the theory of sliding window protocols was unified and, seen as a whole, simplified. The role of transmit and receive windows became more clear and easy both to understand and to teach.

## EFFICIENCY ISSUES

Efficiency is a very important property of communication protocols. It is a measure of the quantity of the service provided by the communication system per quantity of resources used. Most frequently it is defined as the maximum information throughput through the system normalized with the nominal transfer rate of the system; however, other definitions also are possible, heavily depending on which network resources are considered to be more important for a communication system designer (with the above definition, this is the channel capacity). Efficiency depends on the properties of the communication channel, on communication traffic and on the communication protocol as well as its parameters. The communication channel property that most dramatically impacts the efficiency is the channel loss rate. Losses are mainly due to two reasons: errors provoked by the presence of noise and losses that are provoked by the lack of resources in network elements. Although the quality of wireline communication channels has been growing steadily in the past, losses have become again a critical issue with the advent of mobile communication systems where the error rate of radio-based channels may be many orders of magnitude higher than in wireline networks.

It was told already that the efficiency of the selective-repeat protocol is higher than the efficiency of the go-back-N protocol. However, this is evidently true only in case of the efficiency definition mentioned in the previous paragraph where the communication channel capacity is considered as the only communication resource. On the other hand, one must be aware that the selective-repeat protocol requires more buffer memory due to the larger receive window width (almost twice as much as the go-back-N protocol), and, due to the more complex memory management at the receiving side, also more receiver's processing power [2].

The receive window width of the generalised sliding window protocol is larger than that of the go-back-N protocol and smaller than that of the selective-repeat protocol, while the transmit window width is normally the same in both cases (so that the transmit window never becomes full in the lossless case). Along with the increase of the receive window width, the amount of the necessary buffer memory also increases. The protocol efficiency, as defined at the beginning of this section, is also expected to rise with the receive window width. However, both theoretical consideration and experimental results [4] reveal that the efficiency increases at first with the receive window width, but can saturate at some intermediate receive window widths, or even slightly drop towards the selective-repeat case. A typical efficiency versus receive window width dependence is shown in Figure 1.

The dependence of the protocol efficiency on the receive window width that we have just discussed is very important from both engineering and pedagogic viewpoints as it

- gives a protocol designer more freedom in choosing the compromise between the utilisation of channel and memory capacities, and
- can yield equal or even better protocol efficiency while using less memory.

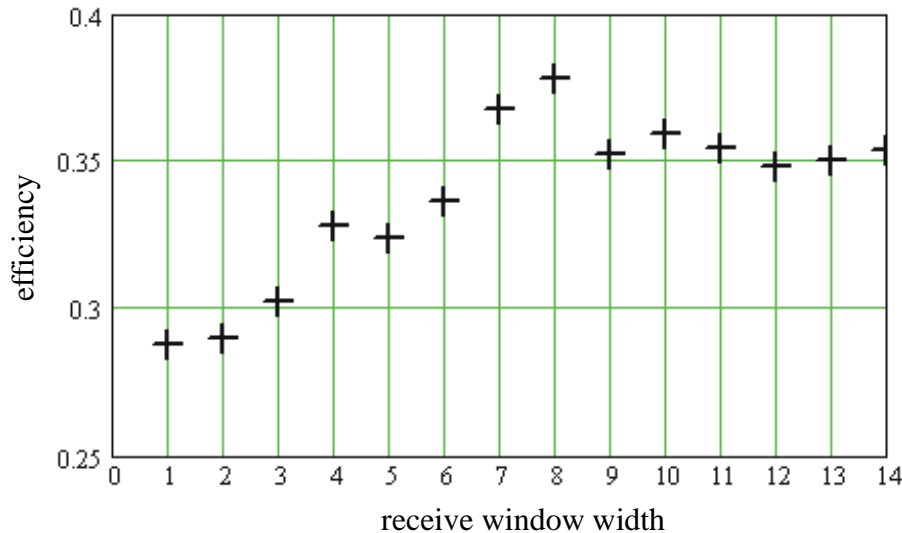


FIGURE 1  
TYPICAL DEPENDENCE OF PROTOCOL EFFICIENCY ON RECEIVE WINDOW WIDTH

## FORMAL PROTOCOL DESCRIPTION

A communication protocol is a set of complex rules that specify how a communication entity must compose, interpret and exchange communication messages with other communication entities. Communication protocol implementation is derived from communication protocol specification. Therefore, the protocol specification must be given clearly and unambiguously, which renders the interoperability of different protocol implementations possible. It is well known [6] that only formal specifications of protocols fulfill these requirements. An introduction into formal protocol specification should therefore be included into any modern higher education communication protocols course.

Several formal languages suitable or even intended for protocol specification exist [7]. One of them, SDL (Specification and Description Language) [5] is standardised, is used by several standardisation bodies, such as ITU (International Telecommunication Union) and ETSI (European Telecommunication Standards Institute), and has been proposed many times (e. g. in [8]) to be used in education, too.

Variants of sliding window protocols were already specified in different formal specification languages. However, the formal specification of the generalised protocol was published in SDL language [4].

## PEDAGOGIC IMPORTANCE OF THE GENERALISED PROTOCOL

The importance of a communication protocols course within a higher education telecommunications programme has already been presented. In this section, we will try to explain why the sliding window protocols and the generalised sliding window protocol in particular are so important to be taught as a part of a communication protocols course.

The importance of the generalised sliding window protocol in the context of communication engineering process is manifold.

Sliding window protocols are very important from theoretical, practical and educational viewpoint. They are typical distributed algorithms which are nontrivial enough to expose the concepts, design and implementation of such algorithms. Furthermore, being the most important data-link layer and transport layer protocols, they are indispensable in the engineering practice. Their educational value stems from their previous properties: theoretically, the understanding of them helps students to grasp the understanding of protocols in general; and practically, most communications engineers will most probably somehow come across sliding window protocols during their professional lifetime.

The generalisation of the sliding window protocols family allows to treat this family of protocols more systematically. By studying the generalised protocol, students can more easily grasp the properties of the family members and the differences between them, as these members are only special cases of the generalised protocol. The explanation of the difference between the classical protocols based solely on the difference in window widths was the weak point of the classical theory, so the role and importance of transmit and receive windows and their widths in the context of the generalised protocol can be explained more easily and cleanly.

The most important property of a communication protocol is its logical correctness. It is therefore extremely important to teach students how to reason about the logical correctness of protocols and how to verify it, either by reasoning, analysis or simulation. Of course, appropriate examples are necessary for that. Without any doubt, the generalised sliding window protocol is very appropriate for this purpose, firstly because of its moderate complexity, and

secondly because one can either reason first about the special cases and then generalise them, or first reason about the general case and then consider also the special cases.

Sliding window protocols can be formally specified at very different levels of complexity. Either special cases (with the stop-and-wait protocol being the simplest) can be separately specified, or the generalised protocol can be first specified, with such specification being later simplified into special cases. Furthermore, even more granularity in specification complexity is offered by the possibility to specify the connection setup phase, connection release phase, or data transfer phase, or all of them.

Efficiency of the sliding window protocols can be analysed either analytically, or using simulation. While the analysis of the lossless case is very simple, the mathematical analysis of the data transfer across a lossy channel requires more probability theory based mathematical tools. In case of simulation-based analysis, only the generalised protocol simulator can be used, as the other protocols are only its special cases. The analysis of the generalised protocol is the most interesting, as its dependence on the receive window width can be studied, with the go-back-N and selective-repeat protocols being its special and extreme cases.

We have already indicated that the choice between the go-back-N, selective-repeat and generalised protocols depends on the compromise between the use of communication channel capacity, buffer memory and computation complexity. Actually, the essential engineering problems are related to solving different compromises. So the sliding window case can provide future engineers with an excellent experience.

## **GENERALISED SLIDING WINDOW PROTOCOL IN A COMMUNICATION PROTOCOLS COURSE**

For the reasons mentioned in the previous Section, the sliding window protocol should and usually does play one of the central roles within a communication protocols course. In this Section, the main points to be explained and emphasised will be presented, along with the special emphasis on the generalised protocol.

Clearly, the principles of automatic repeat request (ARQ) protocols must first be explained, with the necessary protocol mechanisms described and supplementary mechanisms mentioned.

A sliding window protocol is an ARQ protocol with a specific buffer memory management. So the necessity for transmit and receive buffers is to be explained, along with the modulo arithmetic for the sequence counting of packets. The sizes of both buffers are to be discussed and their impact on the protocol operation and retransmit strategy. At this point, the operation of the generalised protocol must be explained and then the relation of its special cases (namely, stop-and-wait, go-back-N and selective-repeat protocols) to it. Timing diagrams must be discussed for typical operation scenarios.

The logical correctness of the protocol must then be discussed. The necessary mechanisms to be used in a logically correct protocol must be explained and usual errors which could lead to a deadlock or a livelock are to be discussed. Possible deadlocks and livelocks must be illustrated with timing diagrams.

Then the protocol must be specified, both informally and formally, to show the difference to the students and to demonstrate how difficult it is to write an unambiguous informal specification of a protocol that is not trivial. For formal specification, an appropriate specification language must be chosen which is standardised, easy to learn and supported by design tools for editing and simulation. According to the author's opinion, the SDL language is currently the best choice, mainly due to its graphic syntax which is very intuitive, good acceptance in industry and good support by design tools, although the UML (Universal Modeling Language) [9] is becoming more and more popular and might prevail in the near future.

Protocol correctness can be verified either by mathematical analysis or using computer-aided simulation tools which search for possible deadlocks and livelocks and can also do the reachability analysis. For undergraduate students, only the simulation seems to be appropriate.

The efficiency of sliding window protocols used on lossless channels is very easy to be analysed; this analysis must be taught, as it clearly shows why the efficiency of the stop-and-wait is low and also explains how the transmit window width of generalised protocol is to be designed to achieve good efficiency. However, the analysis of the sliding window protocol efficiency used on the lossy channel is much more difficult and is usually only approximate; for students (except PhD students), simulation is more appropriate. Wide range of possibilities exist to simulate the dependence of the protocol efficiency on various channel and protocol parameters, such as bit error rate, transmit and receive window widths and timer expiration time. Figure 2 shows an example efficiency versus bit error rate diagram for transmit window width 10 and receive window widths 1, 2, 5 and 10. Such a diagram can be drawn by students based on the simulations they do during their practical work in the lab.

The author has found out that surprisingly many undergraduate students have problems to understand and draw diagrams, especially on logarithmic scale, partly because they got too much used to rely on computer based tools the operation of which they do not understand sufficiently. Therefore, it is a good idea to require undergraduate students to draw some diagrams manually.

Figure 3 overviews the process of sliding window protocol pedagogic presentation.

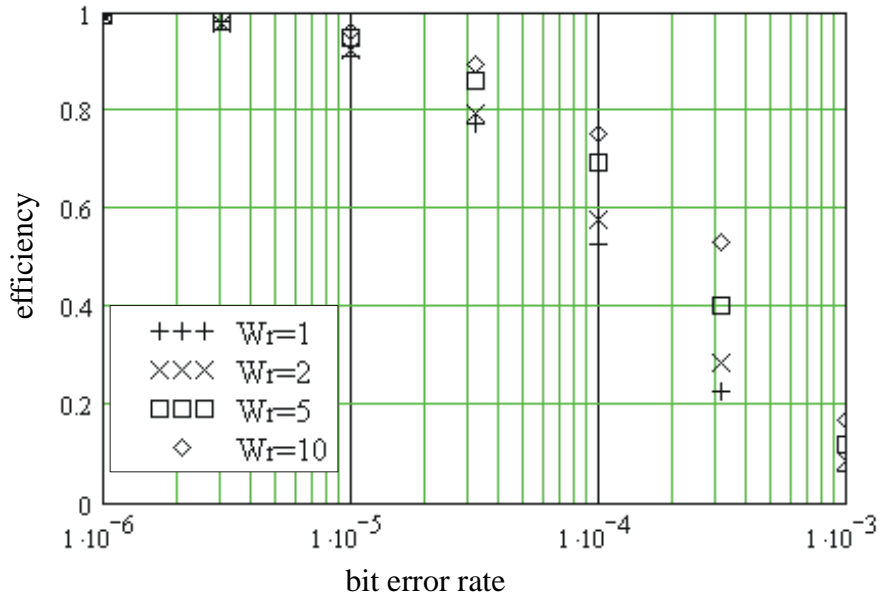


FIGURE 2  
DEPENDENCE OF SLIDING WINDOW PROTOCOL EFFICIENCY ON BIT ERROR RATE WITH RECEIVE WINDOW WIDTH AS PARAMETER

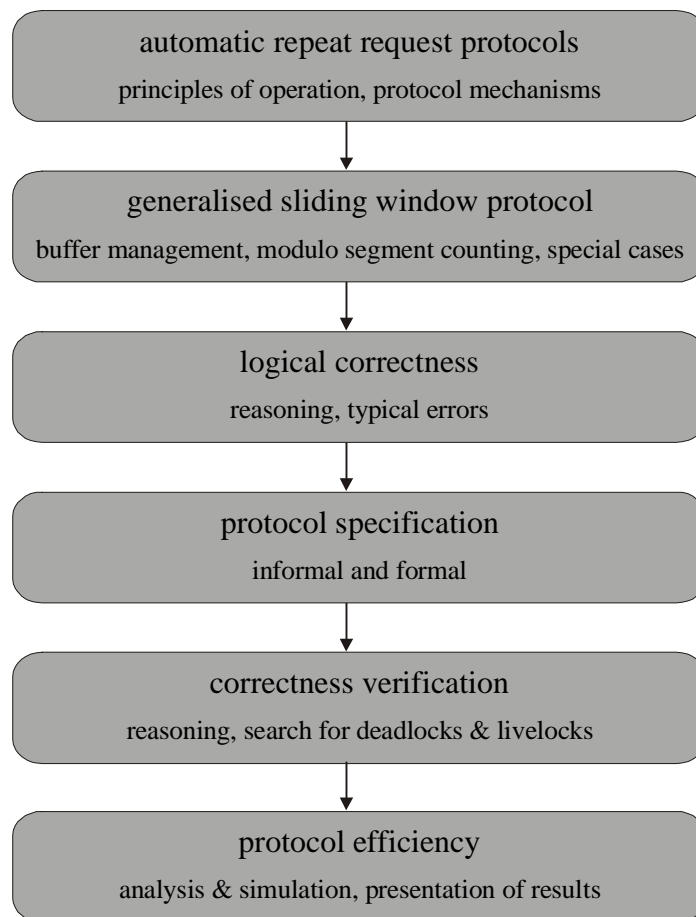


FIGURE 3  
SLIDING WINDOW PROTOCOL PEDAGOGIC PRESENTATION

## CONCLUSION

Sliding window protocols constitute one of the most important communication protocol families both in theory and practice. Hence, they have gained a firm place in virtually all programmes of higher education communication protocols courses. However, three special cases, namely stop-and-wait protocol, go-back-N protocol and selective-repeat protocol are taught, while claiming that they are members of a common protocol family. Indeed, this common family, called generalised sliding window protocols, has been described and formally specified only recently.

In this paper, we argue that the generalised sliding window protocol should be taught in higher education communication protocol courses which would make much easier to explain to students how the protocol operation relates to the transmit and receive window widths. Furthermore, the generalised protocol offers more flexibility in choosing an appropriate compromise between the use of different resources. Then, stop-and-wait, go-back-N and selective-repeat protocols should be explained as the special cases of the generalised protocol.

In the paper, the most important issues relating both the sliding window protocols theory to be taught to students as well as the practical work to be done by students are exposed. The topics described include the protocol operation, informal and formal specification, correctness verification, and efficiency. The knowledge and experience thus gained by students will not be limited to mastering sliding window protocols only, but communication protocols on general.

## ACKNOWLEDGEMENT

This work was supported in part by the research programme Algorithms and optimisation methods in telecommunications of the Ministry of Higher Education, Science and Technology of the Republic of Slovenia.

## REFERENCES

- [1] Stallings, W., *Data and Computer Communications, 8<sup>th</sup> Ed.*, Pearson Education International, Upper Saddle River: 2009
- [2] Tanenbaum, A., S., *Computer Networks, 3<sup>rd</sup> Ed.*, Prentice-Hall International, Upper Saddle River: 1996
- [3] Hercog, D., "Generalised sliding window protocol", *Electron. Lett.*, Vol 38., No 18., 2002, pp. 1067-1068
- [4] Hercog, D., "Generalization of the basic sliding window protocol", *Int. j. commun. syst.*, Vol 18., No 1., 2005, pp. 57-75
- [5] International Telecommunication Union, "Specification and Description Language (SDL)", *ITU-T Z.100*, International Telecommunication Union, Geneva: 2007
- [6] Babich, F., Deotto, L., "Formal Methods for Specification and Analysis of Communication Protocols", *IEEE Communications Surveys & Tutorials*, Vol, 4, No 1., 2002, pp. 2-20
- [7] Lai, R., Jirachiefpattana, A., *Communication Protocol Specification and Verification*, Kluwer Academic Publishers, Boston/Dordrecht/London: 1998
- [8] Hercog, D., "The role of formal languages in engineering education", *ICEE-2004*, Gainesville, Florida, October 17-21, 2004
- [9] Doldi, L., *UML 2 Illustrated, Developing Real-Time & Communications Systems*, TMSO, 2003