A LAPB PROTOCOL COMPUTER-AIDED LEARNING TOOL

Drago Hercog[1]

Abstract *¾ A computer program designed to demonstrate key features of the classic data-link layer communication protocol LAPB is described. The emphasis is placed upon protocol procedures implementing methods for error and flow control, so only the information transfer phase is considered, and the abstract syntax of frames is used. The communication process is presented both graphically and textually and can be considered either as a whole, or from a communicating station's point of view. The program may either be used as a demonstration tool, or a student can interact with different degrees of required experience which leads him to advance his knowledge in a step-by-step manner. The tool is extremely easy to use, and the colour graphics user interface is rich in information and yet self-explanatory. An example learning course using this program is presented. The use of this tool in an undergraduate communication protocol course has yielded good results from both teacher's and students' viewpoints.*

Index Terms ¾ communication protocol, computer-aided learning tool, LAPB, teaching paradigm

INTRODUCTION

The operation of communication systems is inherently complex due to their distributed nature as well as to the large number of states a system can be found in. This complexity extends to communication protocols. It is therefore not an easy task to present protocols to students, especially those at the undergraduate level. This task may be facilitated by carefully preparing the teaching paradigm, as well as allowing students to acquire some hands-on experience by providing for appropriate computer-aided learning tools.

During the last decade, the importance of computer-aided didactic tools to support teaching of communication protocols has become more and more evident. As a result, several programs have been developed and presented in literature. Some of them allowed students to construct their own communication processes to be checked by software for consistency with standards [1]. Others provided for simulated environment in which the protocol code written by students could run [2]-[4]. In this paper we present a computer program developed to illustrate the typical data-link layer protocol LAPB from different viewpoints and to allow students to work interactively with it.

The next section presents some general guidelines for teaching communication protocols. Then the LAPB protocol is briefly overviewed. Program objectives, as well as its limitations and modes of operation are presented before the program design is discussed. Some general guidelines about how the tool is to be used are also given.

TEACHING COMMUNICATION PROTOCOLS

All data communication protocols are based on some general data communication principles and methods. A method is such an abstraction of real-life procedures which is simple enough to be easily understood and general enough to be implemented in different ways, but usually fails to solve real problems due to the lack of details which are indispensable in real life. A general communication method may be implemented with different protocols and even a given protocol may be implemented with different options. So the way from a general method to a protocol or even a working communication program is the way from more abstract to less abstract, or from less details to more details. It is an example of a top-down design process. The learning process must proceed along a similar path. The more abstract general methods must be explained first, and only then some real-life protocols, but still with some degree of abstraction. Peculiar details of specific protocols may be tackled last.

Since several years, the author has been teaching a 30-hour communication protocols course within the frame of the undergraduate telecommunications programme at the University of Ljubljana. Basic methods on which protocols are based, such as automatic error correction and flow control, are first covered. They are further illustrated by detailed discussion of some specific protocols, one of them being LAPB. Then some practical exercises are done by students in the lab which allow them to actively participate in the learning process, using computer learning aids.

A protocol is specified by defining messages that can be exchanged between the protocol entities, as well as how any particular entity should behave. Messages are specified in both their symbolic form (abstract syntax), as well as in the concrete form (transfer syntax). Entity behaviour is specified in terms of rules that determine which messages (usually in their abstract form) may or must be transmitted in any particular situation. Entity behaviour description is based on an entity model that usually consists of inputs and outputs (for receiving/transmitting messages), a processor (for taking decisions), a memory (for storing the state and other values), and timers (for measuring elapsed time). Behaviour rules are the component of protocol specification that is the most difficult to specify unambiguously and without errors.

A communication protocol can be considered from two different viewpoints. One can be interested in the communication process as a whole which is indispensable

[1] Drago Hercog, University of Ljubljana, Faculty of Electrical Engineering, Trzaška 25, SI-1000 Ljubljana, Slovenia, Drago.Hercog@fe.uni-lj.si

when considering protocol validity, performance or protocol design. Graphically, this view is usually presented in the form of time diagrams showing the sequence and/or timing of messages. In this way, different communication scenarios, but of course not all, can be studied. Another possibility is to view a communicating entity (station) as a logic machine with its excitation, response, and state (memory). This view is appropriate for protocol specification and communication program design. Pedagogically, the protocol should be presented from both points of view in an appropriate sequence. As it is difficult to understand the station procedures while ignoring the process as a whole, the communication process must be explained first and only then the station procedures in somewhat more detail.

THE LAPB PROTOCOL

LAPB (Link Access Procedure Balanced) [5] is a classic ITU-T standardised data-link layer protocol. It is connection-oriented and based on Go-Back-N ARQ method for error correction and sliding window for flow control.

One might ask, why just LAPB? An early descendant of the ISO HDLC protocol, it could be considered obsolete today. However, it is simple enough and implements all classic functions of OSI data-link layer. It is also succeeded by several important protocols. E. g., LAPD [6] is very similar to LAPB, with more complex addressing scheme and slightly different, but less efficient timeout recovery mechanism [7]. LAPDm [8] is specialised for use over the radio channel of GSM networks. From the didactic point of view, it is therefore reasonable to explain LAPB first and then only the differences between LAPB and its successors.

LAPB protocol will be described here only in the extent which is necessary to understand the rest of the paper.

During the information transfer phase LAPB frames may be either information (I) frames which transfer the user information, or supervisory frames which serve for control purposes only and include Receive Ready (RR) for positive acknowledgements, Receive Not Ready (RNR) for flow control, and Reject (REJ) for negative acknowledgements. Only I frames include the frame sequence number *N(S)*. All frames include the acknowledgement parameter *N(R)*. The *P/F* parameter indentifies a frame as a P (poll) frame to inquire information from the peer, F (final) frame to answer the enquiry, or none of these. The frames which have been corrupted by errors are discarded.

Information frames are sequentially numbered modulo 8. Acknowledgements for the correctly received frames may be piggybacked onto the information frames, or may be sent in supervisory (RR) frames. Out-of-sequence frames are explicitly rejected with REJ frames. The sender is requested to re-send all the frames from (and including) the first missed one on. If a rejecting frame or the frame which has been resent as a result of a rejection is lost the timer expires and the transmitter has to re-transmit the first unacknowledged frame as a P frame. The sliding window limits the number of frames which have been transmitted but not yet acknowledged. A station may explicitly announce that it is unable to receive new frames with an RNR frame.

The state of a station is defined in terms of several variables. They include *V(S)* which holds the number of the frame to be transmitted next; *V(R)* which indicates the number of the frame the station expects to receive next; *V(A)* which is the number of the next acknowledgement expected; R_{ex} which indicates, if set, a receipt of an out-of-sequence frame that hasn't yet been resolved; similarly, T_{ex} indicates the timeout exception state; the timer status T indicates whether the timer is running or not; finally, the variable X holds the last value of *V(S)* before the timer expired.

LAPB PROGRAM OVERVIEW

Program Objectives

LAPB is a program developed to run on a personal computer with the aim to deepen a student's or a novice communication engineer's understanding of the classical data-link layer communication protocol LAPB. The understanding of Go-back-N ARQ method, sliding window data flow control, and timeout mechanism is emphasised. With this in mind, only the information transfer phase is considered. The program can either be used as a demonstration tool or can be interacted by the user to test his/her knowledge of the protocol. It provides for both graphic and textual presentation of the protocol. The communication process can be viewed either as the whole or from the communicating entity’s point of view. Intended to be used within a short time period by a particular user, its use must be simple and straitforward.

User Interface and Interaction

To be able to run different communication processes, the user can choose several parameters the communication process depends on. They are:

- propagation delay between the two stations,
- supervisory frame transmission time,
- information frame maximum transmission time,
- timer expiration period,
- sliding window width,
- information frame error probability,
- supervisory frame error probability,
- seed for the pseudo random process to generate transmission errors.

Although the time is naturally a real-valued variable and is also treated so internally by the program, the user-input time variables are allowed to be given integer values only; hence all the events of the communication process happen at integer multiples of a basic time unit. This makes the graphic presentation more clear, especially with the time grid drawn in the background. While the supervisory frame transmission time is assigned an exact value, the information frames transmission times are uniformly randomly distributed between the supervisory frame transmission time plus one and the maximum information frame transmission time during the simulation. The timer expiration period is allowed only to be greater than the minimum value which ensures the correct timer operation, or may be assigned zero value to disable the timer. The absence of timer may lead the communication process into a deadlock, which is considered an illustrative case in the light of learning process. Both error probabilities can be input independently which is of course not realistic, but allows more flexibility in user input (e.g., a user may try with the zero supervisory frame error probability first which gives him more chance that the timer will not expire). On general, a user is not expected to input realistic communication parameters; such values should be given which will yield both easy understanding of the communication process and a suitable graphic presentation of it. If, for example, realistic values were given for error probabilities the user might have to wait for quite a long time to see a frame destroyed by an error, not to mention a timeout exception occurrence. The possibility to choose the pseudo random process seed allows one to see different communication processes with the same communication parameters (and therefore 'surprises' for students).

The information on frames is provided in the abstract form

frame_type , *N(S)* , *N(R)* , *P/F*

where *frame_type* may be I (Information frame), RR (Receive Ready frame), or REJ (Reject frame). *N(S)* and *N(R)* can take a value between 0 and 7, and *P/F* can either be P (poll), F (final) or space (none of the above). The *N(S)* parameter is given only for information frames. No details of actual frame construction in terms of fields and bits are given.

Program Limitations

The information transfer phase of LAPB is implemented following the ITU-T recommendation [5], except for the following simplifications. The explicit data flow control by means of RNR frames is not considered. Frames are numbered modulo 8 only, although the recommendation also allows for modulo 128 sequencing. Where the recommendation admits several alternative actions to be taken by a station only one of them is implemented. Stations processing delays are neglected. The link reset is not implemented, as this would mean to exit and then re-enter the information transfer phase. Concerning the communication process, the following was assumed. To avoid ambiguities from the user's point of view, the frame reception events are given precedence over the frame transmission events, should they appear to happen at the same time. The noisy physical layer communication channel as seen from the data-link layer is modelled as a pseudo random process which destroys some frames, and propagates the others without error. The waiting queues at the packet to data-link layer boundaries are assumed to never be empty, thus always having at least one new information packet at hand.

Modes of operation

To allow the student's experience to progress gradually the program can run in one of three basic modes: (1) communication process view without user interaction, (2) communication process view with user interaction, and (3) communicating station behaviour view with user interaction.

The mode (1) is a demonstration of the communication process which scrolls across the screen, with controlled speed as initially required by the user, both graphically (in the form of timing diagram) and textually (in the form of event history, separately for each station).

The mode (2) is similar to the first one in appearance, but requires the user to give his/her opinion on communication decisions which are approved or disapproved by the program. In case of disapproval, the user may either re-enter his/her input or leave the program to continue. In either case, the program insists on its decisions. Correct answers are counted and the user is informed on the success percentage out of total number of answers. The success is calculated for each station separately which allows two students to simultaneously work with the program, each of them interacting with one of the stations.

In this mode the user is offered a menu of three submodes which require different amounts of knowledge and skill. A description of these submodes follows in the order of increasing amount of required experience.

(2a) The user is interrogated on frames and their parameters to be transmitted. In this submode, a user must know how the communication process proceeds as a whole. The resource for his decisions can be either the knowledge of the recent history of communication process, or the current state of the station. In other words, a user may, when deciding about the next frame, consider the communication process as a whole or act like a station.

(2b) The user is interrogated on frames to be transmitted as well as on state changes of stations. To allow him to concentrate on more basic state variables when this is appropriate, the variables used in basic Go-back-N ARQ and sliding window methods are listed first, namely *V(S)*, *V(R)*, *V(A)*, and R_{ex}, and only then the variables T_{ex} and *X* which come into action after the timer has expired. The resource for user's decisions in this submode is the knowledge of the past station state and the event that has occured (reception, transmission or timer expiration). When working in this submode the user must possess a deeper understanding of a station behaviour although he/she still can see the communication process as a whole.

(2c) The user interacts in the timer control in addition to the questions asked in previous submodes. The timer may be passivated, activated, or reactivated (passivated and immediately activated again) when transmitting or receiving a frame or after the timer has expired. This submode requires a thorough understanding of timeout mechanism, in addition to the skills needed in previous submodes.

In the most pretentious mode (3), the peer station and the communication process as a whole are hidden to the user, as is also the history of events. Hence the user can see only the current station state and the most recent event. The degree of interaction is the same as in mode (2c): frame transmissions, state changes and timer are all controlled by him/her. The user's behaviour must therefore emulate that of a real communicating entity, without any additional information available.

PROGRAM DESIGN

General program organization

The LAPB program is built around a discrete-event simulator of the LAPB protocol, which is also the core of a performance simulator and will not be further discussed here. As shown in Figure 1, modules for program mode and communication parameters input, communication process textual and graphic presentation and user interaction are added.

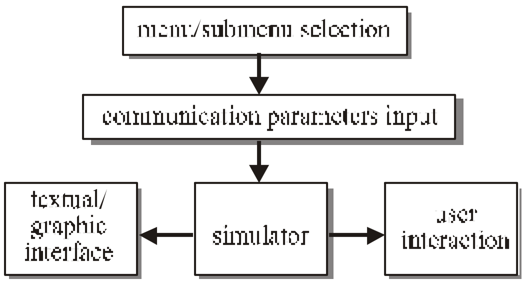


FIGURE 1.
LAPB PROGRAM GENERAL ORGANISATION.

Elements of user interface

Currently, LAPB is a DOS program, but will be converted to Windows environment in the near future. Anyway, the basic elements of its user interface are the same in any environment and comprise (excluding input menus):

- program title,
- communication parameters values,
- communication process graphic presentation (modes 1 and 2 only),
- request/help (all modes, but very simple in mode 1),
- state variables for one or both communicating stations,
- event history for one or both stations (modes 1 and 2), or recent event for one station (mode 3),
- user interaction input for one or both stations (modes 2 and 3),
- success for one or both stations (modes 2 and 3).

In modes 1 and 2, the communication process graphic presentation is considered the most important. It therefore uses light colours on black background, while all other windows use less intensive colours on blue background which yields less contrast. Whenever appropriate, colours are used to encode different types of information, such as different frame types and different degrees of success.

Graphic presentation of communication process

The comunication process as a whole is shown by animating the timing diagram that is used in most textbooks and shown in Figure 2. During the simulation, this diagram is drawn and scrolled, one simulation time unit at a time, over the screen. Consequently, the student can always see a part of the process history, in addition of the current event.

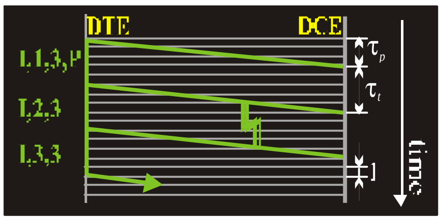


FIGURE 2.
GRAPHIC PRESENTATION OF COMMUNICATION PROCESS.

The positions of both stations (vertical lines) as well as the time grid (horizontal lines) are shown in grey. The frames are drawn in colours which encode frame types: light green for I frame, light red for REJ, and yellow for RR frame. While a frame is being transmitted (which takes transmit time t_t), a coloured line is proceeding down the transmitter station vertical line, and its type and parameters are written besides it. After the transmission has been finished the tail of the frame is shown to propagate across the link as a slanting line headed by an arrow, to reach the receiver boundary after propagation time t_p; at this time the arrow disappears and the frame is received. In case the frame is lost or damaged by an error it is marked by a double lightning sign. As in the recommendation X.25, the two stations are referred to as DTE and DCE, respectively.

To give the illusion that the process is evolving in real time in demonstration mode, a delay of t_d seconds is introduced into the simulator for each time unit of simulation time. The real time delay t_d is given by $t_d = v^{-1}$ where v is the drawing speed in simulated time units per second and is provided by user. In modes 2 and 3 this illusion is lost because the program is being regularly interrupted for user interaction.

Textual information

Additional textual information is also displayed, as appropriate for each mode of operation. The actual communication parameter values, as input by the user, are shown. In modes 1 and 2, an event history shows, for each of the two stations, the list of the 24 most recent events related to the station which is more than is usually shown in the graphic presentation. The event list includes frame transmissions, frame receptions and timer expirations, displayed in green, cyan, and red, respectively, and tagged with labels Xmit, Recv, and Tout. Only the current input event (i.e. frame reception or timer expiration) is shown in mode 3. For both stations (in modes 1 and 2) or for one station only (in mode 3), the state and timer status values are shown and updated after each change.

User's answer validation

The program validates user's answers following the algorithm shown in Figure 3.

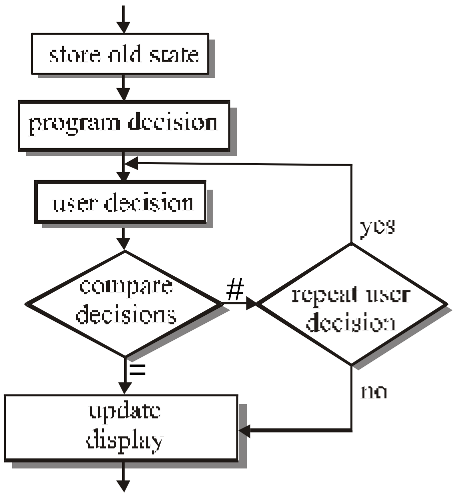


FIGURE 3.
USER'S ANSWER VALIDATION ALGORITHM.

After the program has decided on some action the user is interrogated on his/her opinion. The program then compares its own decisions with those of the user who is notified about the outcome with a consonant (major triad) musical chord in case of success or with a dissonant musical interval (augmented fourth, also known as *diabolo in musica*) in the opposite case. The new success percentage is calculated. It is displayed in green if above 60%, in yellow if between 40% and 60%, and in red otherwise. In case of disagreement the user is offered the choice whether to re-enter his decision or not. Only after the agreement has been achieved or the user gives up (in which case the penalty of five false answers is added to the success percentage) the program's decision is made visible on the screen and the simulation continues.

An example screen

Figure 4 shows the screen (the DOS version of the program) in the moment when the user is requested to input the DTE state changes after a frame has been received by this station.

AN EXAMPLE USE OF LAPB PROGRAM

The program allows the student's knowledge and experience to be gradually improved in successive, more and more demanding interactive tasks. Several different communication processes can be analysed with different degrees of interaction. All of this allows the instructor to choose many different exercises. A possible sequence of communication processes involving more and more protocol features includes the following:

a. error-free interchange of information frames, to get acquaintance with the basics of communication process, as well as its graphical and textual presentation;
b. interchange of frames with low information frame and zero supervisory frame error probabilities, to show how isolated error occurrences are resolved by means of reject frames (instructor has to choose such a random process seed that the information frame which is resent after having been requested by a reject frame is not once more lost which would result in a timeout);
c. a communication process with a higher error rate for both information and supervisory frames, and with the timer shut off which would result in a deadlock, to show the necessity of timer mechanism;
d. the same process as in the preceding example, but with operational timer to prevent the deadlock;
e. a study of the sliding window flow control method with different window widths (including the value 1 which leads one to the stop-and-wait ARQ method, usually used in half-duplex protocols).

Any of these examples can be seen as a demonstration first, and then repeated (possibly with different parameters) in the interactive mode 2, beginning with the frame building submode, followed by user's interaction in state changes, and then, after a student has mastered the protocol pretty well and for exercises d. and e. only, interacting in timer control. Finally, a student can interactively emulate a communicating station in mode 3 of program operation, without any knowledge about the communication process as a whole or about the state or actions of the peer station.

Usually, this whole series of examples is too long to be carried out during a course. However, the instructor can choose an appropriate sequence to fit his/her educational needs within the time frame that is available.

CONCLUSIONS

The LAPB educational program was designed to serve as a learning aid when studying data communication protocols. The tool has been used in a communication protocols undergraduate course. Good results were observed by the author who noticed a substantial improvement of students' understanding of the protocol, as well as by students who declared such a program could help them a lot.

Although the learning aid has, according to the author's opinion, some strong features, especially different degrees of user's interaction and possibility of different views of communication process, it still lacks some features which may be implemented in the future. Let the following lines expose some of these possible future enhancements.

When the recommendation admits several options only one of them is implemented; this is quite adequate in the demonstration mode, but in the interactive mode the program should agree with any user's action which conforms to the recommendation.

For the sake of completeness, the data link setup, reset and disconnection could be included into the program.

During the information transfer phase, each station could be forced, from time to time, into the busy condition; in this way the use of Receive Not Ready (RNR) frame would also be demonstrated.

A program like this can either be used as a stand-alone tool, or be included into a multimedia hypertextbook as a reference illustrating the text. In this case the program would be executed from different places in the text with different parameters, or be used at the end of chapter for several interactive exercises.

REFERENCES

[1] Logan, J., R., King, P. J. B., "PAT: A Protocol Analysis Tutor", *IEEE Trans. Education*, Vol. 36, Aug. 1993, pp. 316-326

[2] Kassabian, D., Albicki, A., "A Protocol Test System for the Study of Sliding Window Protocols on Networked UNIX Computers", *IEEE Trans. Education*, Vol. 38, Nov. 1995, pp. 328-334

[3] King, P. J. B., "dplsim: A Teaching Harness for Data Link Protocols", *IEEE Trans. Education*, Vol. 40, Aug. 1997, pp. 172-178

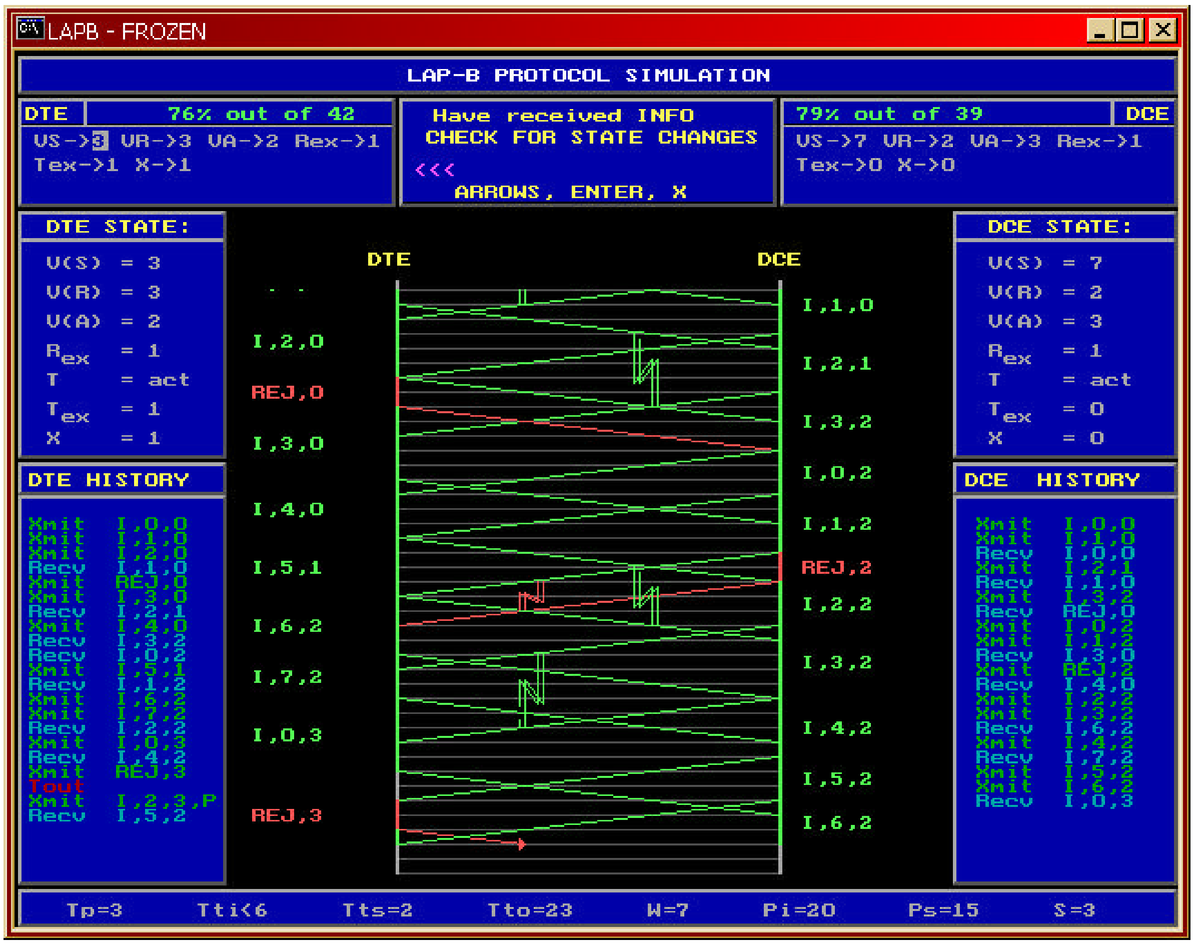
[4] Pullen, J. M., "The Network Workbench: network simulation software for academic investigation of Internet concepts", *Computer Networks*, Vol. 32, 2000, pp. 365-378,

[5] ITU, *ITU-T Recommendation X.25*, International Telecommunication Union, 1993

[6] ITU, *ITU-T Recommendation Q.921*, International Telecommunication Union, 1993

[7] Hercog, D., "Timeout Recovery Strategies in Datalink Layer Protocols at High Bit Error Rates", *Electronics Letters*, Vol. 32, No. 20, Sept. 1996, pp. 1864-1865

[8] ETSI, *European Standard (Telecommunications series) ETSI EN 300 938*, European Telecommunications Institute, 2000


```
LAPB - FROZEN

LAP-B PROTOCOL SIMULATION

DTE      76% out of 42       Have received INFO         79% out of 39       DCE
VS->3 VR->3 VA->2 Rex->1     CHECK FOR STATE CHANGES    VS->7 VR->2 VA->3 Rex->1
Tex->1 X->1                  <<<                        Tex->0 X->0
                               ARROWS, ENTER, X

DTE STATE:                                              DCE STATE:
 V(S)  = 3                   DTE          DCE            V(S)  = 7
 V(R)  = 3                                               V(R)  = 2
 V(A)  = 2                                               V(A)  = 3
 Rex   = 1                                               Rex   = 1
 T     = act                                             T     = act
 Tex   = 1                                               Tex   = 0
 X     = 1                                               X     = 0

        I,2,0                          I,1,0
        REJ,0                          I,2,1
        I,3,0                          I,3,2
        I,4,0                          I,0,2
        I,5,1                          I,1,2
        I,6,2                          REJ,2
        I,7,2                          I,2,2
        I,0,3                          I,3,2
        REJ,3                          I,4,2
                                       I,5,2
                                       I,6,2

DTE HISTORY                                             DCE  HISTORY
Xmit  I,0,0                                             Xmit  I,0,0
Xmit  I,1,0                                             Xmit  I,1,0
Xmit  I,2,0                                             Recv  I,0,0
Recv  I,1,0                                             Xmit  I,2,1
Xmit  REJ,0                                             Recv  I,1,0
Xmit  I,3,0                                             Xmit  I,3,2
Recv  I,2,1                                             Recv  REJ,0
Xmit  I,4,0                                             Xmit  I,0,2
Recv  I,3,2                                             Xmit  I,1,2
Recv  I,0,2                                             Recv  I,3,0
Xmit  I,5,1                                             Xmit  REJ,2
Recv  I,1,2                                             Recv  I,4,0
Xmit  I,6,2                                             Xmit  I,2,2
Xmit  I,7,2                                             Xmit  I,3,2
Recv  I,2,2                                             Recv  I,6,2
Xmit  I,0,3                                             Xmit  I,4,2
Recv  I,4,2                                             Recv  I,7,2
Xmit  REJ,3                                             Xmit  I,5,2
Tout                                                    Xmit  I,6,2
Xmit  I,2,3,P                                           Recv  I,0,3
Recv  I,5,2

Tp=3   Tti<6   Tts=2   Tto=23   W=7   Pi=20   Ps=15   S=3
```

FIGURE 4.
AN EXAMPLE SCREEN.