УДК 623.618

BOOLEAN DERIVATIVES WITH APPLICATION TO NETWORK PROTOCOL TESTING

A. Izotov, V. Nemchenko

Kharkiv National University of Radioelectronics

An approach is suggested for network protocol simulation with application of the finite state machine theory to improve the quality and reliability of information transmission in computer networks.

Problem statement. The problem of building and checking the serviceability and diagnostics of new generation IPv6 protocols is presently in the forefront of research efforts. Design errors resulting in disturbance of conformity – noncompliance of a protocol to its declared specification – increase protocol development time and its cost. This problem is especially acute owing to the huge volume of network protocol replication. Besides, it is often necessary to diagnose already developed and applied protocols whose functioning errors degrade servicing quality and the validity of information transmission in a network [1].

Review of recent research and publications. The Delphi method is currently widespread. It allows solving partially the problem of diagnosing network protocols. However, such an approach requires qualified professionals in network technology areas, this being far from always possible at work sites of Internet users.

Besides, Petri nets are often used for simulating protocols. The drawbacks of this tool as a programming language are that Petri nets have no strict concept of a process, which could be executed on a given processor. A unique sequence of executing a Petri net is also absent because the basic theory provides a language for describing parallel processes. Yet another drawback of the method is that the principles of an object-oriented approach are not applied.

The most promising is the automaton approach to solving the problem of simulating network protocols. However, this approach is being used presently mainly at the stage of protocol application rather than during protocol development and implementation. In addition, the authors limit the potentialities of automaton simulation by reducing it only to representing a protocol as a classical Mealy automaton without employing to full capacity the classical theory of synthesis of microprogram automata [2].

Objective of study. The structure of the life cycle of a network protocol is suggested to be complemented with the phase of building a protocol reference model at its development stage. The reference model can be used to check the results of building a protocol for its conformity to the baseline specification as early as at the initial stage of protocol development. Besides, being based on the classical theory of synthesis of microprogram automata, it is suggested to build test sequences using Boolean derivatives to check the developed protocols.

Basic research materials. The graphical form of representing the specification of the protocol being developed can describe the protocol as a graph of a finite Mealy automaton. For example, the TCP transport protocol can be represented as a Mealy automaton graph (Fig. 1). Table 1 explains the symbols in the transition graph.

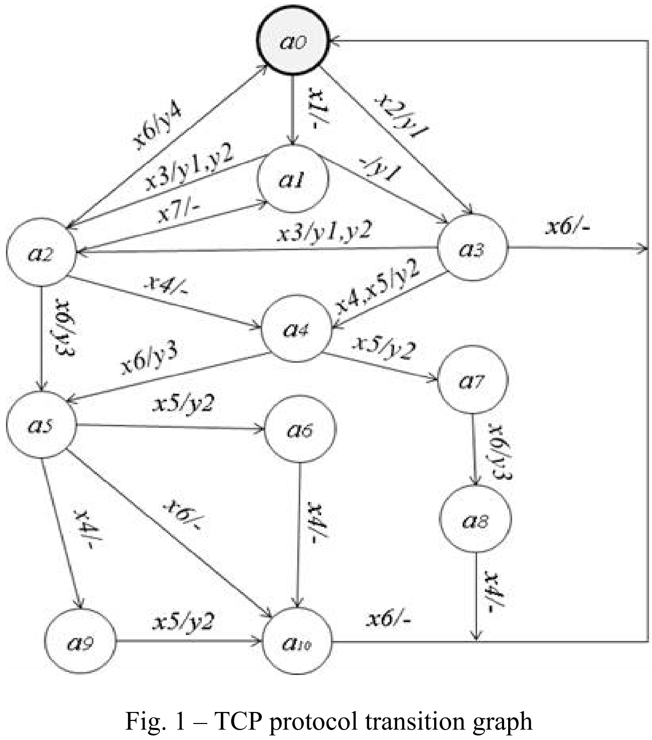

Fig. 1 – TCP protocol transition graph

Table 1 – Meanings of symbols in the TCP protocol graph

Set of states {A}		Set of input instructions {X}		Set of output responses {Y}	
a_0	CLOSED	x_1	Passive Open	y_1	SYN
a_1	LISTEN	x_2	Active Open	y_2	ACK
a_2	SYN_RCVD	x_3	SYN	y_3	FIN
a_3	SYN_SENT	x_4	ACK	y_4	RST
a_4	ESTABLISHED	x_5	FIN		
a_5	FIN_WAIT_1	x_6	close or timeout		
a_6	CLOSING	x_7	RST		
a_7	CLOSE_WAIT				
a_8	LAST_ACK				
a_9	FIN_WAIT_1				
a_{10}	TIME_WAIT				

Based on the known classical theory of synthesis of microprogram automata, we obtain an automaton table of transitions and outputs used to build a system of equations to specify their Boolean functions for automaton transitions and outputs. As an example, let us specify a transition equation for state a_0 (1) and output y_1 (2):

$$a_0 = a_{10}x_6 \vee a_8x_4 \vee a_3x_6 \vee a_2x_6 = x_6(a_2 \vee a_3 \vee a_{10}) \vee a_8. \quad (1)$$

$$y_1 = a_0x_2 \vee a_1 \vee a_1x_2 \vee a_3x_3. \quad (2)$$

Boolean algebra theory includes the concept of a Boolean derivative for both one and several variables. This concept is used widely in the theory of testing digital circuits because the Boolean derivative of function $f(x_1, x_2,..., x_i,..., x_n)$ for variable x_i determines the conditions of activation of the path in the circuit from input x_i to output f(x). The Boolean derivative of function $f(x) = f(x_1, x_2,..., x_n)$ for x_i is known as function (3):

$$df(x) / dxi = f(x1, x2,..., xi,..., xn) \oplus f(x1, x2,..., \overline{xi},..., xn), \quad (3)$$

where $\oplus$ is modulus 2 sum.

Using the principle of activation of paths in the circuit being checked assumes building a test by consecutively determining the check input sets for separate faults. The test is built by combining selected sets. The algorithm for selecting check sets can be presented as follows:

1. Write Boolean function f(x) with variable x_i to be checked for faults $x_i \equiv 1$ and $x_i \equiv 0$.
2. Compute Boolean derivative $df(x)/dx_i$ and reduce the expression obtained to the disjunctive normal form (DNF).
3. Select one of the terms of the DNF obtained.
4. Fault $z \equiv 0$ is checked for an action, at which variables $x_1,...,x_n$ ensure condition $zT = 1$. Similarly, fault $z \equiv 1$ is checked for an action, at which variables $x_1,...,x_n$ ensure condition $\overline{z}T = 1$.

Hence, by activating the path from automaton input to its output, we check for a fault of the given type in the activated path.

Let us consider the method of path activation by example of the above automaton that implements the TCP protocol. Eq. (2) shows that the value at output y_1 is a function of input variables {X} and automaton states {A}. Minimizing the function yields expression (4):

$$y_1 = a_0x_2 \vee a_1 \vee a_1x_2 \vee a_3x_3 = a_0x_2 \vee a_1 \vee a_3x_3. \quad (4)$$

As obvious, function y_1 depends on input variables x_2 and x_3, and on automaton states a_0, a_1, a_3.

Let us find Boolean derivative $df(x) / dx_2$ for y_1 (5).

$$dy_1 / dx_2 = (a_0x_2 \vee a_1 \vee a_3x_3) \oplus (a_0 \overline{x}_2 \vee a_1 \vee a_3x_3) = = a_0\overline{a}_1\overline{a}_3 \vee \overline{a}_1\overline{x}_3. \quad (5)$$

To determine the conditions of activation of the path from input x_2 to output y_1 ($x_2 \rightarrow y_1$), we shall equate the derivative obtained to "1" and define the values of all variables in the given derivative. For this, both terms are first multiplied by $\overline{x}_2$ (to check for fault of type $x_2 \equiv 1$) and then by x_2 (to check for fault $x_2 \equiv 0$). The result shall be presented in cubic form (6).

$$\begin{matrix} x_2 & x_3 & a_0 & a_1 & a_3 & y_1 \\ D & U & 1 & 0 & 0 & D, \end{matrix} \qquad (6)$$

where D = 1/0;
U belongs to set {0, 1}.

Analysis of the test vector obtained shows that we have found a condition, for which a change in variable x_2 changes the value of function y_1, the direction of change being the same. This example shows the similarity of the given approach to building tests for checking network protocols using the theory of path activation in a digital circuit.

By doing similar actions for all output functions (y_1, y_2, y_3, y_4), we obtain test T for checking all paths in an automaton from inputs to outputs (Table 2).

Table 2 – Structure of the TCP protocol check test

T	x_1	x_2	x_3	x_4	x_5	x_6	a_0	a_1	a_2	a_3	a_4	a_9	y_1	y_2	y_3	y_4
t_1	U	D	U	U	U	U	1	0	U	0	U	U	D	-	-	-
t_2	U	0	D	U	U	U	U	0	U	1	U	U	D	-	-	-
t_3	U	D	0	0	0	U	U	1	U	U	U	U	-	D	-	-
t_4	U	0	D	0	0	U	U	U	U	1	U	U	-	D	-	-
t_5	U	0	0	D	1	U	U	U	U	1	0	0	-	D	-	-
t_6	U	0	0	1	D	U	U	U	U	1	0	0	-	D	-	-
t_7	U	0	0	1	D	U	U	U	U	0	1	0	-	D	-	-
t_8	U	0	0	0	D	U	U	U	U	0	0	1	-	D	-	-
t_9	U	U	U	U	U	D	U	U	1	U	1	U	-	-	D	-
t_{10}	U	U	U	U	U	D	U	U	1	U	1	U	-	-	-	D

Conclusions. The material considered shows that modification of the life cycle of network protocols assumes building a reference model of the protocol at the initial stage of its development. This streamlines the process of protocol design and increases the quality of its implementation, and hence, its reliability and quality of data transmission in computer networks. Another important feature of the given method is the possibility of building a check test to check the serviceability of the designed protocol and its conformity to the stated specification.

References

1. Izotov A.S. Automaton approach to testing network protocols / A.S. Izotov // J. Information and Control Systems at Railway Transport (in Ukrainian) - 2012, No. 4, (Supplement). – p. 48-49.

2. Izotov A.S. Modification of the life cycle of network protocols / A.S. Izotov, V.P. Nemchenko, and M.A. Demidchuk // J. Information and Control Systems at Railway Transport (in Ukrainian) - 2014, No. 6. – p. 25-29.

Анотація

ВИКОРИСТАННЯ АПАРАТУ БУЛЕВИХ ПОХІДНИХ ПРИ ТЕСТУВАННІ МЕРЕЖЕВИХ ПРОТОКОЛІВ

Ізотов А. С., Немченко В. П.

Запропоновано підхід до моделювання мережевих протоколів з використанням теорії кінцевих автоматів, що дозволяє підвищити якість і надійність передачі інформації в комп'ютерних мережах.

Аннотация

ИСПОЛЬЗОВАНИЕ АППАРАТА БУЛЕВЫХ ПРОИЗВОДНЫХ ПРИ ТЕСТИРОВАНИИ СЕТЕВЫХ ПРОТОКОЛОВ

Изотов А. С., Немченко В. П.

Предложен подход к моделированию сетевых протоколов с использованием теории конечных автоматов, что позволяет повысить качество и надежность передачи информации в компьютерных сетях.