ВАЛИДАЦИЯ СЕТЕВЫХ ПРОТОКОЛОВ НА ОСНОВЕ КОНЕЧНО-АВТОМАТНОЙ МОДЕЛИ

Сивков С.А. Email: Sivkov1135@scientifictext.ru

Сивков Сергей Александрович – магистрант,
Институт информационных технологий и компьютерных систем
Севастопольский государственный университет,
г. Севастополь

Аннотация: *предлагается конечно-автоматная модель для исследования характеристик, влияющих на безопасность работы протоколов распределенных технических систем. Исходными данными для исследования является первичная спецификация протокола, описанная таблицей переходов взаимодействия объектов распределенных систем. Разработана формальная модель представления протокола и основные требования, предъявляемые к протоколу. В качестве метода проверки характеристик безопасности протокола выбран анализ дерева достижимых глобальных состояний.*
Ключевые слова: *валидация, несостоятельность протокола, протоколы информационного обмена, спецификация протокола, расширенный конечный автомат, дерево достижимых глобальных состояний, конечно-автоматная модель TCP протокола.*

VALIDATION OF NETWORK PROTOCOLS BASED ON THE FINITE-AUTOMATIC MODEL

Sivkov S.A.

Sivkov Sergey Aleksandrovich – master,
INSTITUTE OF INFORMATION TECHNOLOGIES AND COMPUTER SYSTEMS
SEVASTOPOL STATE UNIVERSITY,
SEVASTOPOL

Abstract: *a finite-automatic model is proposed for the study of characteristics that affect the safety of the protocols of distributed technical systems. The initial data for the study is the primary specification of the protocol, described by the table of transitions between the interaction of distributed system objects. A formal model for presenting the protocol and basic requirements for the protocol have been developed. As the method of checking the security characteristics of the protocol, an analysis of the tree of attainable global states is chosen.*
Keywords: *validation, protocol insolvency,* i*nformation exchange protocols, protocol specification, extended finite automaton, tree of reachable global states, finite automaton model of TCP protocol.*

УДК 004.057.4

В современном мире большой объем личной, коммерческой и прочей информации передается через открытые локальные и глобальные сети. Стремительный рост исследований в области сетевых и информационных технологий в последней четверти XX века привел к развитию направления, которое связано с разработкой протоколов информационного обмена. Важной задачей является задача исследования и определения безопасности современных протоколов, а также разработка новых безопасных протоколов.

Анализ безопасности протоколов информационного обмена состоит в обнаружении возможных несостоятельностей в протоколах.

В настоящее время обрели развитие математические методы формального анализа безопасности протоколов. Основу таких методов составляют процедуры формального описания протокола с последующей верификацией работы модели протокола. Большинство известных методов формального анализа ориентированы только на доказательство традиционных требований безопасности (например, секретности и аутентичности). Как показывает реальная практика, обнаружение несостоятельностей в протоколах может происходить и происходит спустя длительное время после опубликования, разработки и внедрения протоколов. Возможно, что новые несостоятельности и угрозы безопасности обнаруживаются в протоколах, которые ранее уже были проанализированы с использованием одного или нескольких методов [1].

Наиболее подходящей абстрактной моделью протокола является расширенный конечный автомат [2], который в дальнейшем будем называть протокольным автоматом (PA).

PA определяет взаимодействие, как минимум, двух процессов. Описание такой модели можно представить в виде пары взаимодействующих протокольных автоматов PA_i и PA_j:

PA_i::<{{+m_{ji}}∪{-m_{ij}}},{P_i^k},P_i^0,$\gamma_i = \gamma_i(P_i^k, \pm m_{ij}^k)$,$F_i$>;

PA_j::<{{+m_{ij}}∪{-m_{ji}}},{P_j^r},P_j^0,$\gamma_j = \gamma_j(P_j^r, \pm m_{ji}^r)$,$F_j$>.

Здесь введены следующие обозначения:

{+m_{ji}}({+m_{ij}}) – сообщения, принимаемые PA_i (PA_j) от PA_j (PA_i);

{-m_{ij}}({+m_{ji}}) – сообщения, посылаемые PA_i (PA_j) к PA_j (PA_i);

{P_i^k} – множество процессов (k=1,2,…,K), реализующих сервис протокола PA_i;

{P_j^r} – множество процессов (r=1,2,…,R), реализующих сервис протокола PA_j;

P_i^0 (P_j^0) – начальные (активизирующие) процессы PA_i (PA_j);

$\gamma_i(\gamma_j)$ – функции выходов, определяющие каким процессом PA_i (PA_j) формируется сообщение посылка к PA_j (PA_i);

F_i (F_j) – совокупность процессов PA_i (PA_j), завершающих формирование сервиса.

Каждое состояние РА интерпретируется как внутренний процесс обработки или формирования сообщений. Передача сообщений осуществляется через коммуникационную среду. Модель этой среды представляет собой совокупность очередей с известной системой обслуживания. Например, если два процесса обмениваются сообщениями через полудуплексный канал и дисциплина обслуживания очереди сообщений C_{ij} (передача от PA_i к PA_j) и C_{ji} (передача от PA_j к PA_i) определяется как FIFO (First Input - First Output), то описание модели канала можно представить тремя правилами функционирования:

- посылка сообщения

$$P_i^{'} = \gamma_i(P_i, -m);\ C_{ij}^{'} = C_{ij}.m;\ P_j^{'} = \gamma_j(P_j, -m);\ C_{ji}^{'} = C_{ji}.m;$$

- приём сообщения

$$P_i^{'} = \gamma_i(P_i, +m);\ C_{ij} = m.C_{ij}^{'};\ P_j^{'} = \gamma_j(P_j, +m);\ C_{ji} = m.C_{ji}^{'};$$

- внутреннее событие без посылки, либо приёма сообщений (таймаут)

$$P_i^{'} = \gamma_i(P_i, \emptyset);\ C_{ij}^{'} = C_{ij};\ P_j^{'} = \gamma_j(P_j, \emptyset);\ C_{ji}^{'} = C_{ji}.$$

Здесь $C_{ij}^{'}$, $P_i^{'}$, $C_{ji}^{'}$, $P_j^{'}$ - состояния каналов и РА после завершения события.

Предложенная модель позволяет определять текущее состояние системы взаимодействия двух объектов РС как (P_i, C_{ij}, P_j, C_{ji}), которое будем называть глобальным состоянием.

Модель глобального состояния РС, состоящей из N взаимодействующих объектов, можно представить квадратной матрицей $(PC)_{NxN}$ диагональные элементы которой определяют локальные состояния PA_i (i= 1, 2, ..., N), а на пересечении i-й строки и j-го столбца фиксируется состояние канала связи C_{ij}.

Глобальное состояние всей системы можно также наглядно представить в виде матрицы, первая строка которой описывает состояние очередей сообщений и подтверждений, передаваемых от объекта P_i к объекту P_j, а вторая – состояние аналогичных очередей сообщений и подтверждений, следующих в обратном направлении:

$$\begin{bmatrix} P_i & \rightarrow & C_{ij} \\ \uparrow & & \downarrow \\ C_{ji} & \leftarrow & P_j \end{bmatrix}.$$

Для реализации задачи валидации протоколов используется метод анализа дерева достижимых глобальных состояний (ДДГС) [2].

К преимуществам метода следует отнести удобную графическую форму представления и достаточно простой способ автоматизации процесса анализа. Созданные на основе этого метода автоматизированные системы применялись для исследования ряда реальных протоколов [3]. Основным недостатком метода является быстрый рост числа глобальных состояний по мере роста сложности протоколов. Известно несколько разновидностей данного подхода: метод перебора, метод диалоговых матриц, метод фазовых диаграмм, метод «прилегающих» состояний и метод совместных путей.

По окончании генерации новых глобальных состояний, создается граф, узлы которого являются частичными глобальными состоянии ми. а дуги — возможными переходами между этими состояниями. Анализ такого графа дает возможность проверить следующие свойства протокола:

- потерю или неспецифицированный прием входного события (свойство 2). Канал К (J. I) содержит событие, а в автомате I нет перехода из текущего состояния, помеченного приемом этого события;
- тупиковое состояние (свойство 1). Для всех I (1≤I≤N) текущее состояние G(I) не имеет переходов, помеченных выходными или внутренними событиями, а все каналы пусты;
- переполнение среды (свойство 5). Возникает в том случае, если среди всех автоматов найдется хотя бы один такой, в котором имеется переход из текущего состояния, помеченный выходным событием в канал К(I. J), и число событий в этом канале равно Cap(I, J);
- избыточная спецификация (свойство 4). Возникает в том случае, если в каком-либо автомате найдется переход, который ни разу не был использован при построении графа достижимых глобальных состояний.

Корневой вершиной ДДГС является начальное глобальное состояние ($P_i^0,\ C_{ij}=\emptyset,\ P_j^0,\ C_{ji}=\emptyset$). Переход на новую (текущую) вершину определяется событием, реализуемым текущим процессом $P_i^k(P_j^r)$. Валидация выполняется путем обхода ДДГС от корневой вершины до листьев дерева. Если обнаруживается несоответствие, то вершины данной ветви, расположенные ниже, уже не анализируется.

Рассмотрим пример валидации системы «клиент-сервер» на основе протокола TCP.

В технологии разработки протоколов существует набор свойств, обязательный для любых протоколов. Поэтому проверка корректности протокола должна обеспечивать доказательство того, что спецификация протокола обладает этими свойствами. Свойства эти следующие:

1. Отсутствие статических блокировок. Это значит, что в протоколе не существует такого состояния или набора состояний, из которых нет переходов в другие состояния.

S=$\begin{bmatrix}2 & \lambda\\ \lambda & 1\end{bmatrix}$.

Каналы C_{ij} и C_{ji} пусты, а автоматы PA_i и PA_j, находятся соответственно в состояниях s_2^i и s_1^j, из которых не предусмотрен переход по посылочным операциям.

2. Полнота, то есть протокол обеспечивает реакцию на все возможные входные сообщения (отсутствуют ошибки неспецифицированных приемов).

В таблице переходов, автомат PA_j, находясь в любом состоянии, всегда может принимать сообщения m, находящиеся в канале C_{ij}. Т.е. не должно быть таких ситуаций, когда автомат PA_j не знает, как реагировать на входное сообщение m.

3. Однозначность соответствия состоянии, то есть отсутствие таких протокольных объектов, у которых одно состояние может сосуществовать с несколькими различными состояниями какого-либо другого объекта.

В дереве достижимых глобальных состояний отсутствуют глобальные состояния, которые уже повторялись на ранее рассмотренных уровнях.

Не может глобальное состояние S=$\begin{bmatrix}2 & \lambda\\ \lambda & 1\end{bmatrix}$ на первом уровне дерева повторяться на другом уровне, например – третьем S=$\begin{bmatrix}2 & \lambda\\ \lambda & 1\end{bmatrix}$.

4. Отсутствие избыточности, то есть в спецификации протокола нет непоступающих сообщений и невыполняемых действий.

Не может быть такого глобального состояния S=$\begin{bmatrix}2 & \lambda\\ \lambda & 5\end{bmatrix}$, если в рассматриваемых автоматах нет 5 состояния.

5. Ограниченность, которая означает, что во время функционирования протокола число сообщений в каждом канале между протокольными объектами не превышает определенного значения, называемого емкостью канала.

Например, при емкости канала равной 2, не может быть такого глобального состояния, в канале которого находится более 2 сообщений - S=$\begin{bmatrix}2 & 1*3*3\\ \lambda & 5\end{bmatrix}$ или S=$\begin{bmatrix}2 & \lambda\\ 2*1*3*4 & 5\end{bmatrix}$.

6. Отсутствие динамических блокировок. В протоколе отсутствует бесконечный цикл функционирования, при котором не производится полезная работа. Различают динамические блокировки, выход из которых логически невозможен, и динамические блокировки, являющиеся следствием определенных временных характеристик протокола, например, темпа обмена сообщениями.

7. Завершаемость (развитие), т.е. протокол всегда достигает конечного (терминального) состояния. Для циклических протоколов это свойство несколько видоизменяется. Эти протоколы должны обладать свойством развития, которое состоит в том, что протокол достигает своего начального состояния.

Если начальное глобальное состояние S=$\begin{bmatrix}0 & \lambda\\ \lambda & 0\end{bmatrix}$, то и конечное состояние должно быть S=$\begin{bmatrix}0 & \lambda\\ \lambda & 0\end{bmatrix}$, где λ – может содержать любые m.

8. Самосинхронизация (восстановление после непредвиденной ситуации). Это свойство подразумевает, что после возникновения ненормальной ситуации протокол за конечное время восстановит свое корректное функционирование.

При соблюдении этих свойств в исследуемом протоколе, можно считать отсутствие несостоятельности у рассматриваемого протокола.

Проиллюстрируем взаимодействие сетевых объектов «Клиент - Сервер» на основе TCP протокола. Из существующей диаграммы состояний протокола, выделяют следующие состояния соединения TCP - LISTEN, SYN_SENT, SYN_RECEIVED, ESTABLISHED, FIN_WAIT_1, FIN_WAIT_2, CLOSE_WAIT, CLOSING, LAST_ACK, TIME_WAIT и фиктивное состояние CLOSED (фиктивно, так как представляет состояние, когда уже нет TCB и соединения) [4]. Ниже кратко описаны все эти состояния.

LISTEN - ожидание запроса на соединение от любого удаленного TCP и порта.

SYN_SENT - ожидание соответствующего запроса на соединение после передачи своего запроса.

SYN_RECEIVED - ожидание подтверждения соединения после передачи и приема запросов на организацию соединения.

ESTABLISHED - соединение действует и принятые данные могут быть доставлены пользователю. Это нормальное состояние для процесса обмена данными через соединение.

FIN_WAIT_1 - ожидание запроса на разрыв соединения от удаленного TCP или подтверждения для ранее переданного запроса на разрыв соединения.

FIN_WAIT_2 - ожидание запроса на разрыв соединения от удаленного TCP.

CLOSE_WAIT - ожидание запроса на разрыв соединения от локального пользователя.

CLOSING - ожидание подтверждения от удаленного TCP для запроса на разрыв соединения.

LAST_ACK - ожидание подтверждения для запроса на разрыв соединения, переданного удаленному TCP (это подтверждение включается в запрос на разрыв соединения от удаленной стороны).

TIME_WAIT - ожидание пока пройдет достаточно времени, чтобы быть уверенным в приеме удаленным TCP подтверждения для его запроса на разрыв соединения.

CLOSED - соединения уже нет (разорвано).

Соединение TCP переходит от одного состояния к другому в ответ на события, к числу которых относятся пользовательские вызовы OPEN, SEND, RECEIVE, CLOSE, ABORT и STATUS, входящие сегменты (в частности те, которые включают флаги SYN, ACK, RST, FIN) и тайм-ауты.

Для проверки несостоятельности протокола, представим работу TCP системой взаимодействующих расширенных автоматов. Представим графическую интерпретацию взаимодействующих PA_1 и PA_2 через протокол TCP (рисунок 1) [4].

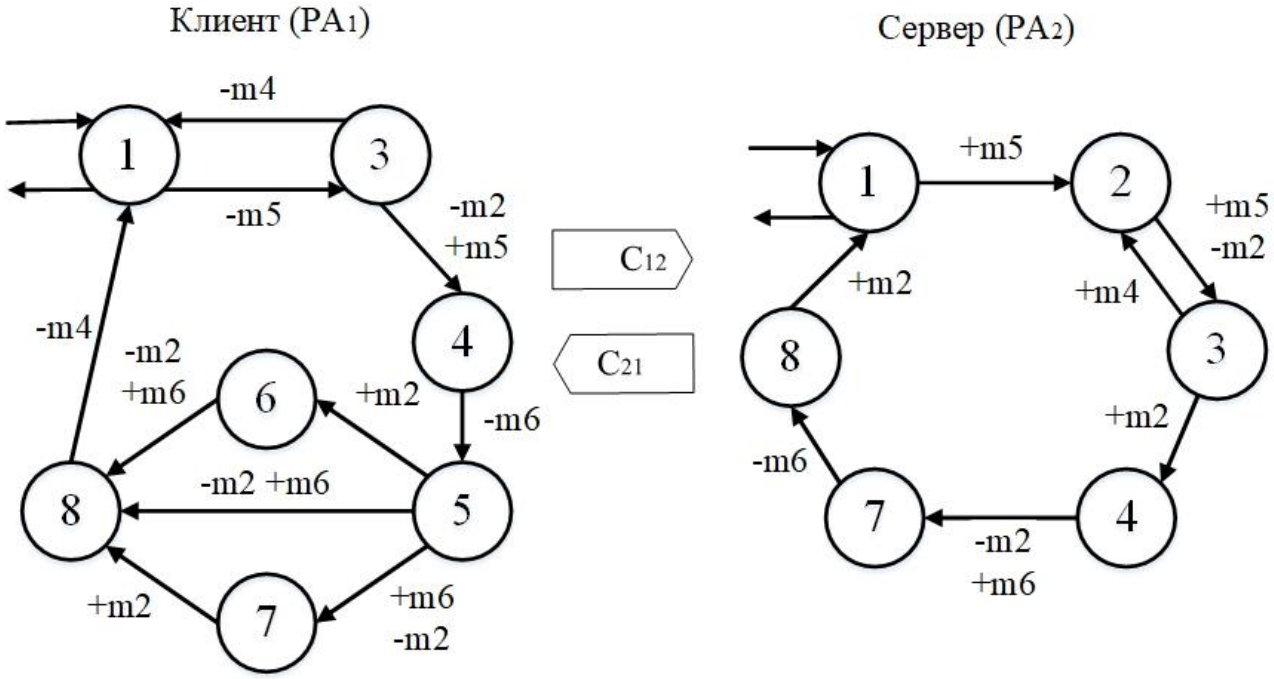

Рис. 1. Исходное представление системы протокольного взаимодействия «Клиент-Сервер» через TCP

Описание протокольных автоматов представлено в таблице 1.

Таблица 1. Описание состояний протокольных автоматов

Состояние	Клиент	Сервер
1	Начальное состояние. CLOSED	Начальное состояние. CLOSED
2	-	LISTEN
3	SYN_SENT	SYN_RECEIVED
4	ESTABLISHED	ESTABLISHED
5	FIN_WAIT_1	-
6	FIN_WAIT_2	-
7	CLOSING	CLOSE_WAIT
8	TIME_WAIT	LAST_ACK

Возможные сообщения при работе протокола:

m1: URG: указывает на срочность приема;
m2: ACK: подтверждение доставки данных;
m3: PSH: выталкивание данных;
m4: RST: сброс соединения;
m5: SYN: синхронизация порядковых номеров;
m6: FIN: у отправителя больше нет данных.
Построение дерева ДДГС представлено на рисунке 2.

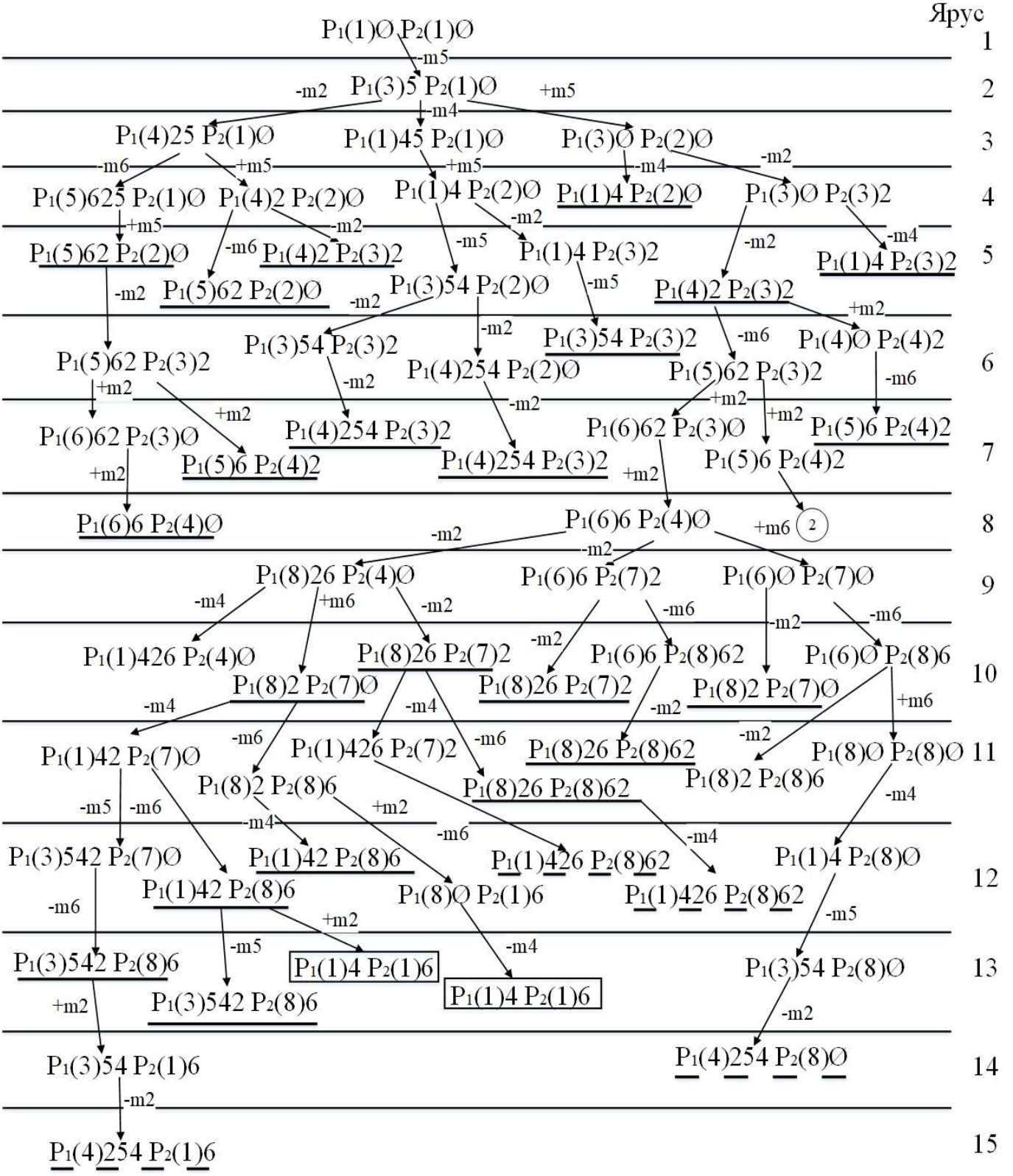

Рис. 2. ДДГС

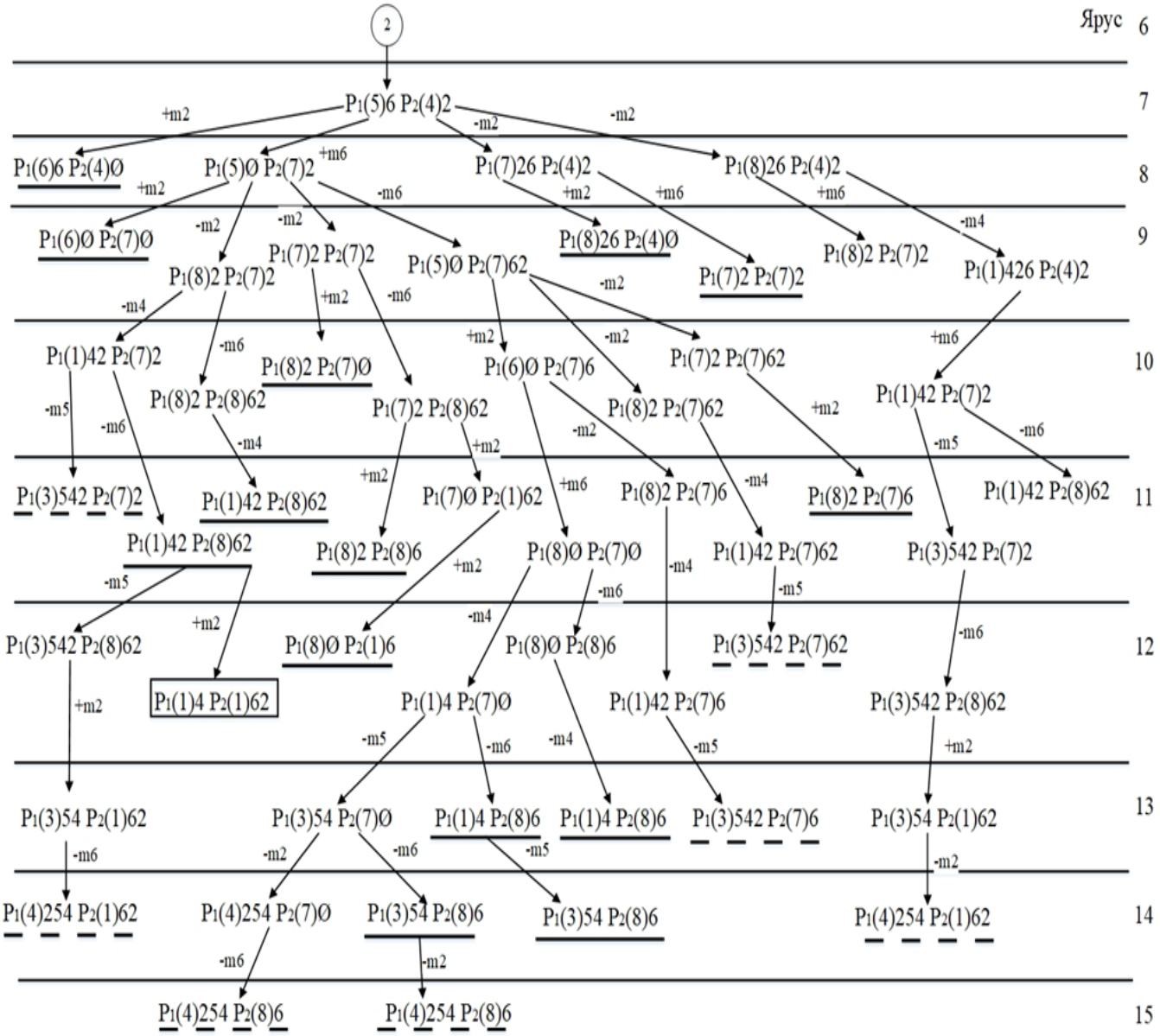

Рис. 3. Продолжение ДДГС

При анализе ДДГС можно выделить следующие состояния: в ДДГС есть вершины (на ярусах 11–15), их которых не выполняется передача/прием сообщений из-за соблюдения требований, ограничивающих ёмкость канала. При увеличении объёма очереди увеличится и глубина дерева, но подобные состояния останутся. На ярусах 4–14 есть состояния, из которых осуществляется переход в другие состояния, которые находятся на других ярусах, что свидетельствует о возможности возникновения циклов, но в данном случае циклы не бесполезные – система всё равно переходит в конечные состояния. В результате построения ДДГС, получено 3 конечных состояния, которые соответствуют диаграмме работы TCP.

Предложена система для исследования несостоятельности сетевых протоколов на основе взаимодействия систем, представленных конечно-автоматной моделью. Перечислены характеристики протоколов, несоблюдение которых говорит о несостоятельности. Исследовано конечно-автоматное представление протокола TCP. Построено ДДГС. Время обработки дерева равно 15 условным единицам. Выявлены несостоятельности рассмотренного протокола – зацикливание и переполнение емкости канала.

Список литературы / References

1. *Lowe G.* Breaking and fixing the Needham-Schroeder public-key protocol using FDR. // Software - Concepts and Tools. 17: Pp. 93-102, 1996.

2. *Апраксин Ю.К.* Моделирование поведения взаимодействующих объектов распределённых систем / Ю.К.Апраксин // АСУ и приборы автоматики. Харьков, 1999. Вып. 110. С. 3-6.
3. Experience with Formal Specification Using an Extended State Transition Model / G.V. Bochmann, E.Cerny, V.Gague E.A. // IEEE Trans., 1982. Vol. COM-30. № 12. P. 2506-2512.
4. *Сивков С.А.* Исследование несостоятельности сетевых протоколов на основе конечно-автоматной модели / С.А. Сивков, Ю.К. Апраксин // Материалы внутривузовской студен. науч.-техн. конф., Севастополь: СевГУ, 2017. С. 108.