

УДК 623.618

ИЗОТОВ А.С., аспирант,

НЕМЧЕНКО В.П., к.т.н., доцент,

ДЕМИДЧУК М.А., магистр (Харьковский национальный университет радиоэлектроники)

Модификация жизненного цикла сетевых протоколов

Рассмотрена проблема проверки конформности реализации сетевых протоколов. Показана необходимость модернизации жизненного цикла протокола. Дан подход к моделированию сетевых протоколов с использованием теории конечных автоматов. Предложенный подход может быть использован для решения задачи построения автоматизированных диагностических систем проверки сетевых протоколов.

Ключевые слова: протокол TCP, конформность, тестирование, конечный автомат, моделирование, жизненный цикл.

Анализ предметной области и постановка задачи

Анализируя состояние дел в области сетевых телекоммуникаций, мы видим большое количество и разнообразие сетевых протоколов, используемых сегодня в локальных и глобальных сетях. Более того, сейчас происходит переход от использования стека протоколов TCP/IP версия 4 к стеку TCP/IP версия 6. Это ведет, с одной стороны, к модификации существующих и созданию новых протоколов, с другой стороны, к необходимости проверки на совместимость новых реализаций протоколов с существующими протоколами.

Одной из актуальных задач на этапе создания программного продукта (в том числе и сетевых протоколов), является задача тестирования его перед внедрением в «производство» поскольку тестирование на этапе производства и сопровождения ПО требует значительно больших затрат по сравнению с затратами на тестирование на этапе проектирования и разработки [1]. Особую актуальность решению данной задачи придает тот факт, что разрабатываемое сетевое ПО имеет чрезвычайно широкое применение в масштабах всей глобальной мировой сети.

В литературе отмечается, что для того, чтобы успешно выполнить проект, объект проектирования должен быть, прежде всего, правильно и адекватно описан, то есть необходимо построить полноценные и функциональные информационные модели объекта проектирования. Как известно, стандарт ISO/IEC 12207 описывает структуру процессов жизненного цикла (ЖЦ) программного обеспечения (ПО), но не конкретизирует в деталях, как реализовать или выполнить действия и задачи, включенные в эти процессы.

Жизненный цикл программного обеспечения

Сегодня существует несколько типовых моделей ЖЦ ПО. Среди них отметим каскадную и циклическую модели. Для наглядности предлагается использовать модифицированную каскадную модель ЖЦ, которая предполагает последовательную реализацию отдельных этапов проектирования и создания ПО с возвратом после каждого этапа на любой из предыдущих этапов в случае нарушения правильности выполнения проекта. На рис. 1 показана упрощенная структура жизненного цикла сетевого протокола.



Рис. 1. Упрощенная структура ЖЦ сетевого протокола

Наряду с выполнением этапов, представленных на рис. 1, предлагается выполнение новых дополнительных этапов: 1 - создание эталонной модели протокола; 2 - анализ конформности протокола; 3 - моделирование ошибок протокола.

Введение данных этапов предполагает еще на этапе проектирования и создания протоколов производить их тестирование на правильность реализации, что позволит значительно снизить вероятность некорректной реализации задекларированной спецификации протоколов. Рис. 2 демонстрирует изменения, внесенные в «классический» ЖЦ сетевого протокола (рис. 1).

© А.С. Изотов, В.П. Немченко, М.А. Демидчук, 2014

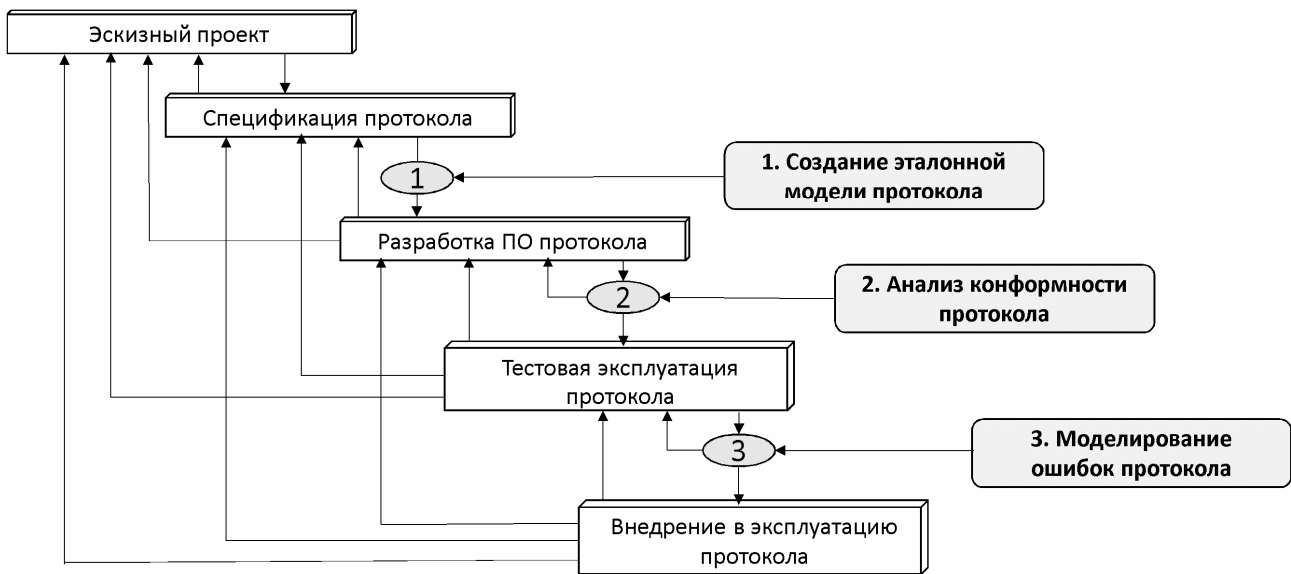


Рис. 2. Модифицированная каскадная модель ЖЦ протокола

Отметим, что введение этапа создания эталонной модели протокола создает условия для реализации остальных предлагаемых вновь вводимых этапов ЖЦ; этап анализа конформности протокола и этап моделирования ошибок протокола.

Этап анализа конформности протокола в случае получения отрицательного результата позволяет возвратиться к этапу разработки ПО с целью внесения определенных изменений в него.

Моделирование ошибок протокола выявленных в процессе тестовой эксплуатации последнего позволяет возвратиться на этап разработки ПО протокола с

целью изменения структуры ПО для устранения выявленной ошибки.

Как известно, каскадная модель имеет ряд недостатков, которые усложняют ее использование. На смену данному типу модели ЖЦ ПО в последнее время приходит модель спирального типа. На рис. 3 показано внедрение новых этапов, предложенных выше, в типовую спиральную модель ЖЦ протокола. Нумерация вновь вводимых этапов та же, что и на рис. 2.

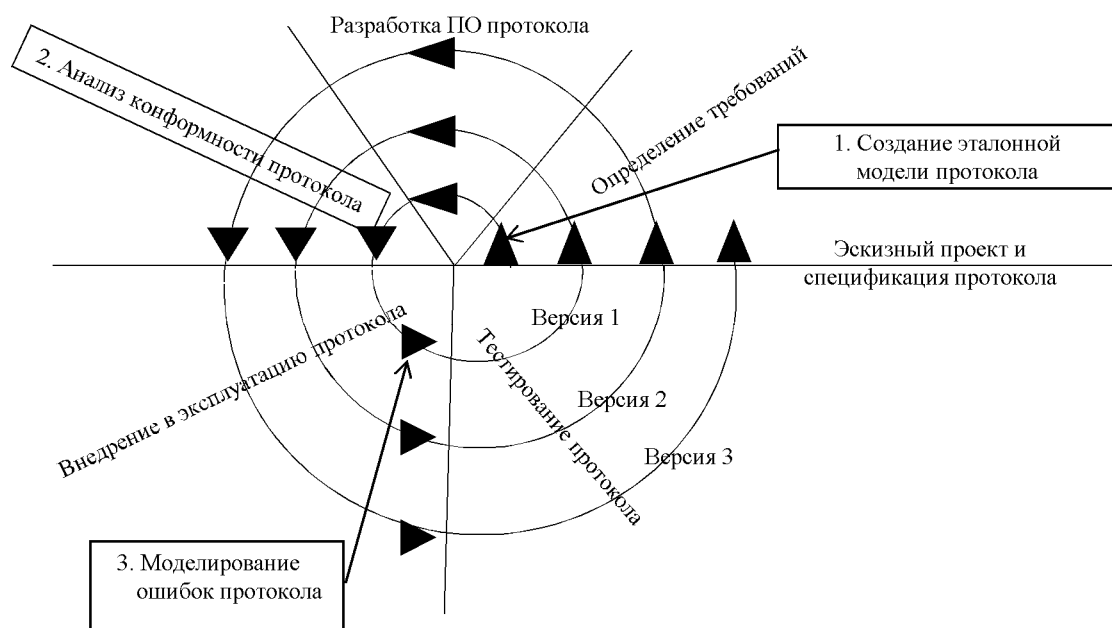


Рис. 3. Модифицированная спиральная модель ЖЦ протокола

Эталонная модель протокола

Исходными данными для эталонной модели протокола является его спецификация, задекларированная при разработке протокола. В качестве примера авторы взяли спецификацию сетевого протокола транспортного уровня – TCP четвертой версии [2].

Существует несколько подходов к моделированию сетевых протоколов, среди них выделим два основных: метод сетей Петри и метод, использующий теорию моделирования конечных автоматов (Finite-State Machine - FSM) [3, 4]. Оба они в общем случае адекватно описывают поведение протокола. Анализ формы спецификаций сетевых протоколов однозначно свидетельствует о том, что наиболее адекватно описывает поведение сетевых протоколов модель автомата Мили.

В ходе исследований была создана модель протокола TCP в среде программирования Active HDL, с помощью которой проанализирована возможность тестирования протокола на его конформность заданной спецификации. Также было проведено моделирование некоторых возможных ошибок в работе данного протокола.

Для осуществления проверки на выполнение графом условий достижимости вершин и состояний был создан проект, состоящий из одного графа, и проведен обход всех его дуг и вершин.

Анализ конформности протокола

В качестве тестов полученной модели на адекватность была выполнена проверка возможности обхода всех состояний и всех переходов графа.

Результат проверки возможности обхода всех состояний графа: длина теста - 16 циклов; покрытие состояний - 100.00%; покрытие переходов - 42.42%. Результат проверки возможности обхода всех переходов графа: длина теста - 42 цикла; покрытие состояний - 100.00%; покрытие переходов - 100.00%. Как видим, результаты тестирования свидетельствуют, что представленный граф выполняет требования достижимости состояний и переходов и может использоваться в качестве основы для дальнейшего использования в системе тестирования протоколов.

Кроме того, была проведена проверка базовой функциональности модели протокола. Для этого используя диаграмму обмена данными было промоделировано установление TCP-соединения как со стороны клиента, так и со стороны сервера. При этом на устройстве клиента при установлении соединения используется обратная схема и при корректном закрытии соединения на стороне клиента TCP-протокол переходит в состояние ожидания закрытия соединения. После получения завершающего пакета ACK (Acknowledgement) от сервера, протокол

на клиентском устройстве переходит в состояние ожидания для возможной пересылки последнего ACK-пакета на сервер и по истечении таймера переходит в начальное состояние. Результаты тестирования показывают полную функциональность модели протокола.

Также исследования показали, что использование одинарного графа протокола для дальнейшего тестирования конформности является проблематичным, так как он в силу своей структуры не позволяет показать текущее состояние TCP протокола в заданный момент времени на клиентском и серверном устройстве одновременно.

Для решения данной проблемы авторами было введено понятие гипермодели (двойной) модели протокола. Введем в среду тестирования еще один граф протокола TCP, который является идентичным тому, который мы тестировали ранее. Это позволяет наглядно смоделировать нестандартные ситуации и реакцию как клиентского, так и серверного приложения на них. Однако прямое соединение входов и выходов двух конечных TCP автоматов не позволит напрямую регулировать их работу, поэтому введем дополнительный модуль – перехватчик (*англ.* – *Interceptor*), на входы которого поступают выходные сигналы от обоих TCP-автоматов, а на его выходах будут моделироваться управляющие сигналы для этих автоматов. Таким образом, данный модуль можно использовать для установки автоматов в требуемые для тестирования состояния и моделирования разнообразных нестандартных ситуаций или неисправностей (рис. 4).

Помимо управляющих схем, данный модуль также эмулирует работу таймеров, посылку данных и опознает процесс разрыва соединения, которые предусмотрены в реальных TCP устройствах на аппаратном уровне, но были упразднены в графической модели.

При соединении данного модуля с двумя графами получим итоговую тестовую среду. Данная среда с одной стороны использует модель работы TCP протокола в виде графа, а именно его программной интерпретации, а с другой стороны сама является моделью работы двух независимых TCP устройств.

Моделирование ошибок протокола

Авторами была проведена симуляция установки TCP - соединения с использованием гипермодели. В частности, была проверена возможность эмулировать такие нестандартные ситуации как синхронное открытие и закрытие TCP соединения (*англ.* - *Simultaneous Open* и *Simultaneous Close*), предусмотренные спецификацией TCP протокола. Также был промоделирован случай одновременного закрытия соединения. В спецификации TCP-протокола такая ситуация возможна и также описана, что

позволяет избежать задержку из-за нестандартных хоть и маловероятных ситуаций.

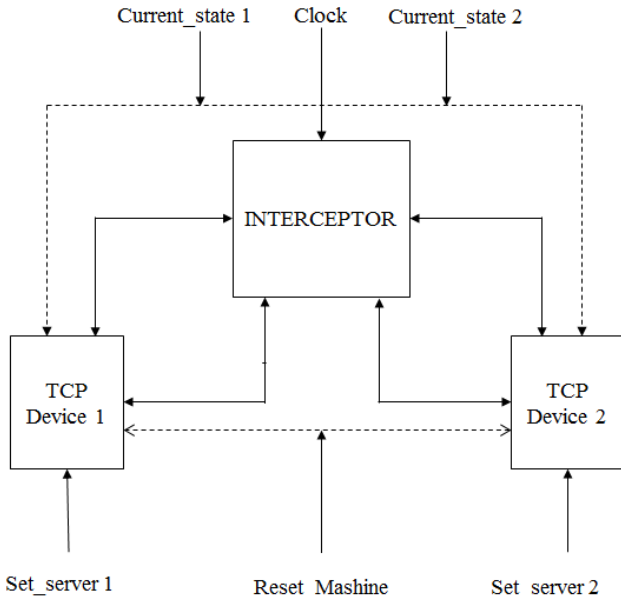


Рис. 4. Гипермодель протокола TCP

Кроме того, с помощью полученной модели было проведено тестирование протокола в условиях получения ошибочных данных, в аварийных ситуациях и при неисправностях оборудования. Для этого осуществлялось моделирование наиболее распространенных ошибок в работе TCP-протокола таких как «разрыв соединения», «нерабочий таймер» и т. п.

Полученные результаты показали, что гипермодель TCP - протокола адекватно моделирует его поведение и пригодна для ее использования с целью анализа конформности протокола его спецификации, а также при моделировании возможных ошибок в работе протокола.

Выводы

Введенный авторами в жизненный цикл протокола процесс построения эталонного протокола заключается в построении эталонного автомата, реализующего протокол. Это позволяет разработчику еще на этапе проектирования протокола осуществить его проверку на соответствие задекларированной спецификации, что значительно упрощает и удешевляет сам процесс проектирования.

Также мы имеем возможность проанализировать поведение протокола на практике при внесении в его работу некоторых возмущающих воздействий (ошибок) путем анализа получаемых выходных реакций автомата. Такой подход может быть положен в основу не только метода верификации сетевых

протоколов, но и определения типа ошибки, возникающей в процессе их эксплуатации. Предложенный в работе подход к проверке конформности протоколов является универсальным и может использоваться при проектировании новых сетевых протоколов.

Предлагаемый метод моделирования ошибок протокола дает возможность построения автоматизированных диагностических систем проверки сетевых протоколов.

Литература

1. Тестирование ПО [Электронный ресурс] / Полезная информация для тестировщиков - Режим доступа : [www/ URL: http://testingworld.ru/category/](http://www.testingworld.ru/category/) - 2014.
2. Изотов А. С. Анализ конформности на этапе проектирования сетевых протоколов [Текст] / А. С. Изотов, В. П. Немченко. - Проблемы информационных технологий. Вестник Херсонского национального технического университета, № 01(15). – Херсон, 2014. - с. 206-210.
3. Немченко В. П. Автоматный подход к тестированию сетевых протоколов [Текст] / В. П. Немченко. А. С. Изотов. - Вестник Херсонского национального технического университета, № 2(38). – Херсон, 2010. - с. 272–275.
4. Баранов С. И. Синтез микропрограммных автоматов [Текст] / С.И. Баранов. - Л.: Энергия, 1979. – 232 с.

А. С. Изотов, В. П. Немченко, М. О. Демидчук. Модифікація життєвого циклу мережєвих протоколів. Розглянуто проблему перевірки конформності реалізації мережєвих протоколів. Показано необхідність модернізації життєвого циклу протоколу. Дано підхід до моделювання мережєвих протоколів з використанням теорії кінцевих автоматів. Запропонований підхід може бути використано для вирішення завдання побудови автоматизованих діагностичних систем перевірки мережєвих протоколів.

Ключові слова: протокол TCP, конформність, тестування, кінцевий автомат, моделювання, життєвий цикл.

A. S. Izotov, V. P. Nemchenko, M. O. Demydchuk.

Modification of network protocols life cycle. The problem of conformity verification of a network protocols realization has been considered. The necessity to modernize a protocol life-cycle is shown. An approach to network protocols modeling using the theory of finite-state mashines has been given. The proposed approach allows testing the protocol at the its design stage. Protocol reference model makes it possible to identify the type of the projected protocol errors. Method of the designed protocols conformity checkout proposed in this paper is a universal method; it can be used in the building of automated diagnostic systems of network protocols checkout to generate the test-case sequence.

Key words: TCP protocol, conformity, testing, finite-state machine, simulation, life-cycle.

Рецензент д.т.н., профессор Кривуля Г.Ф.
(Харьковский национальный университет
радиоэлектроники)

Поступила 10.09.2014г.