

УДК 004.057.4

*Изотов А.С.**Харьковский национальный университет радиоэлектроники*

Начало 90-х годов прошлого века было ознаменовано бурным развитием сетевых технологий, вызванным сложностью в маршрутизации и проблемами с агрегированием адресов. Все это привело к необходимости разработки стека протоколов нового поколения.

Интенсивная разработка новых сетевых протоколов в последнее время приводит к острой необходимости тестирования новых протоколов как на соответствие техническим спецификациям, так и на возможность их корректного взаимодействия между собой и с уже существующими протоколами. Решение данных вопросов приобретает особую актуальность в связи с переходом на новое поколение сетевых протоколов стека TCP/IP версии 6. Возможны различные пути решения этой задачи: подготовка тестов экспертами – специалистами в области сетевых технологий; разработка ручного метода построения тестов; создание специализированной автоматизированной системы генерации тестовых последовательностей.

На наш взгляд, третья возможность представляет наибольший интерес, поскольку первая требует участия высококвалифицированных экспертов в процессе разработки тестов, вторая – также требует участия в разработке квалифицированных специалистов – тестировщиков.

Основной задачей настоящей работы является разработка общей структуры системы тестирования сетевых протоколов с проработкой отдельных составных частей системы.

Предлагается система тестирования сетевых протоколов, приведенная на рис. 1 в состав которой входят следующие блоки: TG (Test Generator) – генератор тестовых наборов; TC (Test Converter) – преобразователь тестовых наборов, приводящий их к единому стандартизированному виду; CORE (Main functional block) – основной функциональный блок, ядро системы; RA (Results Analyzer) – анализатор результатов тестирования.; RVR (Results View Rendering) – блок визуализации результатов. В качестве входных интерфейсов можно назвать: tech-docs, dev-notes – техническая документация, заметки разработчиков; protocol's model – логическая модель протокола; protocol's specification – спецификация протокола.

Кроме перечисленных блоков система содержит следующие типы потоков данных: 1- обработчик черновых материалов на базе UniTESK; 2- таблица тестовых входных воздействий; 3- нормализованные наборы тестов; 4- графическая модель протокола; 5- результаты тестирования в виде таблицы значений; 6- заявленные нагрузки и характеристики в виде таблицы значений; 7- единая таблица сравнений результатов с ожидаемыми значениями.

Проанализируем структуру, приведенную на рисунке 1. Отметим, что ее создание вызвано необходимостью иметь универсальный механизм, который позволит провести проверку правильности функционирования разрабатываемого теста, как на этапе разработки, так и после ее окончания основываясь на утвержденной спецификации.

Первым звеном этого механизма является блок генерации тестов (TG), входные значения которого получены основываясь на заметках разработчиков на этапе производства и на утвержденной спецификации после выпуска последней стабильной версии протокола. Выходные данные этого блока

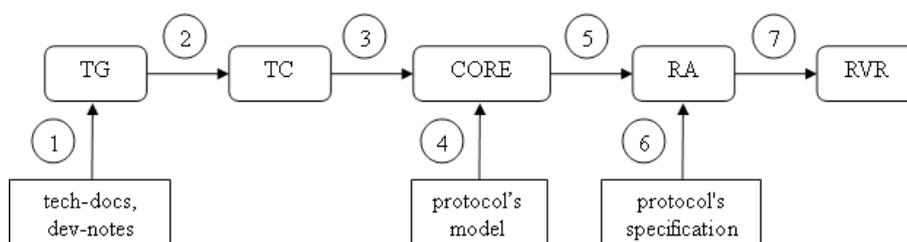


Рисунок 1. Общая структура системы тестирования

необходимо представить в виде структурированной таблицы критических значений с параметром нагрузки.

Вторым блоком модели есть анализатор выходных значений генератора (TC), который должен интерпретировать их в единый формат для покрытия всех логических элементов протокола.

Исполнительным звеном системы тестирования (ядром системы) является третий блок (CORE). Это блок с двумя потоками входных данных: приведенных к единому формату, то есть тестовые наборы и модель протокола. Данные, которые получены после анализа спецификации покрывают модель протокола и результате мы получаем набор выходных значений тестирования классифицированных по заявленным нагрузкам.

Анализ полученных результатов тестирования осуществляет конечный блок (RA). Результаты, полученные при испытании, сравниваются с заявленными техническими характеристиками и определяются возможные дефекты, которые могут возникнуть при реальных нагрузках после введения протокола в массовую эксплуатацию.

Так же планируется использовать дополнительный блок для отображения результатов тестирования в виде графиков и диаграмм (RVR), которые необходимо наложить на заявленные в спецификации характеристики.

На этапе концептуального проектирования системы тестирования сетевых протоколов необходимо решить ряд проблемных вопросов, первым из которых является — в каком виде представлять входные данные первого блока (заметки разработчиков и черновые варианты документаций)? Как мы знаем, стандартным форматом нормативной документации протоколов Интернет являются документы RFC (Request for Comment). Требования в этих документах изложены на английском языке и представляют собой неформальный текст, описывающий желаемое поведение системы. В рамках технологии UniTESK для формальной записи требований используются спецификационные расширения языков программирования – Java или C.

Технология UniTESK пронизывает весь жизненный цикл разработки программного обеспечения от сбора и анализа требований до его сопровождения. Основанная на опыте реальных промышленных проектов в компаниях со сложившейся культурой разработки, технология UniTESK не требует для внедрения коренной перестройки процессов. Она легко сочетается с другими подходами к тестированию и обеспечению качества, обогащая их возможности и обогащаясь при этом сама. Все элементы технологии UniTESK подчиняются требованию прослеживаемости требований от этапа анализа потребностей пользователей до выпуска конечного продукта, в нашем случае сетевого протокола. Технология представляет собой сочетание хорошо зарекомендовавших себя техник, которые могут применяться в различных комбинациях, взаимно сочетаясь, и усиливая друг друга. Это делает технологию гибкой и настраиваемой под существующие процессы разработки на всех этапах жизненного цикла разработки протокола от сбора и анализа требований до сопровождения.

Запись неформальных требований нормативной документации на формальном языке представляет собой модель протокола. В подходе UniTESK формальная модель протокола строится в терминах конечных автоматов. Переходы между состояниями могут задаваться как в явном виде, так и в неявном. В случае явного задания перехода модель содержит алгоритм вычисления следующего состояния и реакции протокола; неявное задание перехода представляет собой предикат, который накладывает ограничения на допустимые конечные состояния и реакции протокола.

Вторым вопросом, требующим решения на этапе концептуального проектирования системы тестирования сетевых протоколов, является представление модели тестируемого протокола, основываясь на одной из его наиболее стабильных версий.

В настоящее время на практике часто используется автоматный подход к моделированию сетевых структур, а именно, используется теория конечных автоматов и теория графов. Такой подход является оправданным, поскольку всякая спецификация протокола может быть достаточно легко представлена в виде содержательного графа автомата, который описывается следующими множествами: множеством состояний автомата, множеством входных событий, множеством выходных событий, а также множеством выходных функций и множеством переходов автомата. Затем

полученная модель представляется в виде графа, узлы которого являются состояниями системы, дуги помечаются входными событиями, которые переводят автомат из одного состояния в другое, а также выходными событиями.

При моделировании сетевых протоколов, представленных в виде спецификации, можно использовать как теорию конечных автоматов, так и метод сетей Петри. Анализ предлагаемых моделей показал их адекватность и возможность использования в системе генерации тестовых последовательностей для проверки сетевых протоколов на их соответствие спецификации. Оба подхода могут быть использованы для решения рассматриваемой задачи. Выбор одного из методов целесообразно производить после анализа конкретной спецификации. Он зависит от наличия соответствующего программного симулятора и от сложности спецификации.

Таким образом, предлагаемая схема тестирования сетевых протоколов может применяться на любом этапе их проектирования и разработки, и должна максимально эффективно выявлять возможные дефекты, способствуя активному развитию и безопасному внедрению новых решений в области интернет-индустрии.

- [1] Fujiwara S., Bochmann G. V., Khendek F., Amalou M., Ghedamsi A. Test selection based on finite state models // Universite de Montreal. Canada. 2006.
- [2] Немченко В.П., Изотов А.С. Построение системы генерации тестовых последовательностей для сетевых протоколов: Харьков - Інформаційно-керуючі системи на залізничному транспорті, №4, 2011. – с. 73-80.
- [3] I. B. Bourdonov, A. S. Kossatchev, V. V. Kuliamin, A. K. Petrenko. UniTesK Test Suite Architecture.// Proceedings of the International Symposium of Formal Methods Europe, July 22-24, 2002, Copenhagen, Denmark; LNCS 2391, 2002, pp. 121-152.
- [4] Utting, M., Legeard, B.: Practical Model-Based Testing: A Tools Approach. Morgan Kaufmann, San Francisco (2007)