

УДК 519.713

М.В. ЖИГУЛИН, С.А. ПРОКОПЕНКО

СИНТЕЗ ТЕСТОВ ДЛЯ ПРОТОКОЛА TFTP НА ОСНОВЕ АВТОМАТНОЙ МОДЕЛИ С ТАЙМАУТАМИ

Приведены экспериментальные результаты по синтезу проверяющих тестов для тестирования доступных реализаций протокола TFTP на основе автоматной модели с таймаутами.

Ключевые слова: автоматная модель, таймаут.

1. Синтез тестов на основе обхода графа переходов

Краткое описание протокола TFTP (Trivial File Transfer Protocol) взято из [1]. TFTP – простой протокол, который предназначен для чтения файлов (сообщений) с удаленного сервера и/или передачи файлов (сообщений) на удаленный сервер. Каждая передача начинается с запроса на чтение или запись файла, который является запросом на установление соединения. Если сервер удовлетворяет запрос, то соединение устанавливается и файл пересылается как последовательность блоков, каждый из которых не более 512 байт. Пакет длины менее 512 байт означает конец передачи. В случае потери пакета в сети абонент (после некоторого таймаута) может заново передать последний пакет, который может быть как пакетом с данными, так и подтверждением. В каждый момент отправитель должен иметь в точности один пакет для ретрансляции, поскольку предыдущие подтверждения гарантируют, что предыдущие пакеты уже получены. Большинство ошибок вызывают закрытие соединения, и ошибка определяется посылкой пакета ERROR. Этот пакет не подтверждается и не повторяется, т.е. TFTP сервер или пользователь может закончить сеанс передачи после отправки такого сообщения, при этом на другом конце этот факт может быть не известным. Для обнаружения такого прекращения сеанса в случае потери пакета ERROR используется система таймаутов. Ошибки инициируются событиями трех типов: нет возможности удовлетворить запрос (не найден файл, запрещен доступ, нет такого пользователя), получен пакет, который нельзя объяснить задержкой или повторением в сети (некорректно сформированный пакет, например) или потерей доступа к необходимым ресурсам (диск переполнен или запрещен доступ). Протокол распознает только одну ошибку, которая не вызывает закрытие соединения, а именно некорректное имя порта, откуда отправлен пакет. В этом случае пакет ERROR теряется. Протокол поддерживает пять типов пакетов: 1) запрос на чтение (RRQ); 2) запрос на запись (WRQ); 3) данные (DATA); 4) подтверждение (ACK); 5) ошибка (ERROR). Пакеты RRQ и WRQ отправляются клиентом для установления соединения с сервером для чтения или записи файлов.

Соединение устанавливается после отправления пакета RRQ (WRQ), на что сервер отвечает отправкой первого пакета с данными DATA_1 (или пакетом подтверждения ACK_0 на запрос WRQ). Каждый пакет DATA имеет свой номер, начиная с 1. Соответственно при подтверждении отправляется сообщение ACK с тем же номером. Таким образом, когда сервер посылает ответ ACK_0 на запрос о чтении файла, последнее только означает, что запрос подтвержден, и сервер готов передавать данные. Если сервер отвечает сообщением ERROR, то соединение закрывается, и клиент должен снова отправить запрос на чтение или запись файлов. Когда соединение установлено, может начаться передача данных (с подтверждениями) между клиентом и сервером. Во время передачи пакеты могут потеряться, т.е. пакет не обязательно доставляется получателю, и чтобы повысить надежность, используется система таймаутов. До отправления следующего пакета должно быть получено подтверждение о получении предыдущего. Если подтверждение не получено в течение определенного времени, пакет посылается еще раз, пока число попыток не превысит допустимое число. В последнем случае соединение закрывается. Вместо подтверждения может быть отправлен пакет ERROR. После отправления подтверждения на последний пакет получатель может закрыть соединение.

На рис. 1 представлен автомат с таймаутами, описывающий поведение TFTP. Для того чтобы получить более простое описание, мы тестируем только часть поведения, которая отвечает за получение файлов от сервера (запрос RRQ). Мы также предполагаем, что в передаваемой последовательности содержится не более 3 пакетов, не допускается повторная передача пакета и время ожидания пакета равно 3 с, и вносим соответствующие изменения в доступные реализации этого протокола. Как показывают проведенные нами эксперименты, даже при таких ограничениях тест, построенный по временному автомату, обнаружил ошибки в протокольных реализациях.

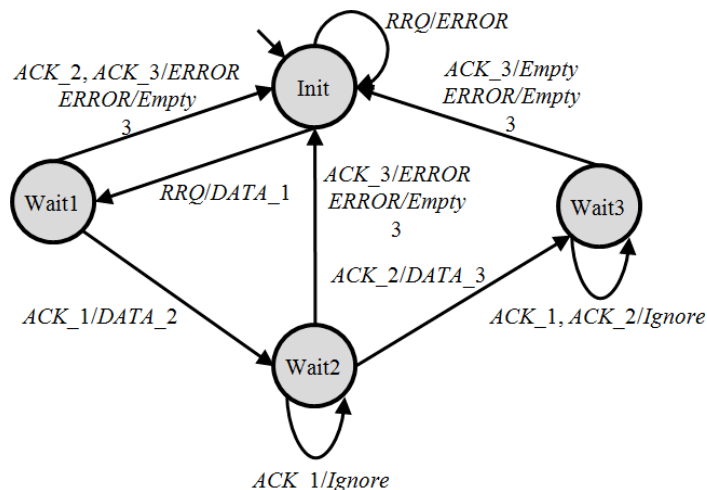


Рис. 1. Автомат с таймаутами для протокола TFTP

По временному автомату на рис. 1 был построен соответствующий конечный автомат, как описано в работе [2]. Построенный автомат получился недетерминированным и частичным. Соответственно были выбраны его детерминированные подавтоматы таким образом, чтобы при совмещении обходов графов переходов подавтоматов получился обход графа переходов исходного автомата. Общая длина построенного теста равна 674 временных входных символов.

По временному автомату на рис. 1 был построен соответствующий конечный автомат, как описано в работе [2]. Построенный автомат получился недетерминированным и частичным. Соответственно были выбраны его детерминированные подавтоматы таким образом, чтобы при совмещении обходов графов переходов подавтоматов получился обход графа переходов исходного автомата. Общая длина построенного теста равна 674 временных входных символов.

2. Экспериментальные результаты

Построенный, как описано в предыдущем разделе, проверяющий тест был использован для проверки реализаций протокола TFTP, которые находятся в открытом доступе. Тестирование проводилось для следующих реализаций: класс **TFTPServer** из библиотеки **commons-net-2.0.0**, разработчик Apache [3]; сервер **atftpd** для Linux, разработчик Jean-Pierre Lefebvre [4].

Для обеих программ было заявлено, что они поддерживают протокол RFC1350 [1]. В результате тестирования были найдены следующие несоответствия. Класс из библиотеки **commons-net** игнорирует пакеты подтверждения с номерами пакетов, которые не были еще отправлены, когда в соответствии с протоколом (RFC1350 п. 2) данные пакеты должны восприниматься как некорректно сформированные и отправляться соответствующая ошибка. В сервере **atftp** была выявлена серьезная ошибка, поскольку при подаче подтверждающего пакета с любым номером блока, сервер отвечал пакетом с данными с номером следующего блока, т.е. если на вход подавалась последовательность (RRQ,0)(ACK_3,0), то на выходе получали DATA_1, DATA_4, что никак не соответствует спецификации протокола (RFC1350 п. 2 и п. 5).

Таким образом, можно сделать вывод, что тесты, построенные обходом графа переходов соответствующего конечного автомата, являются достаточно качественными и дают возможность обнаруживать ошибки в реализациях современных протоколов.

СПИСОК ЛИТЕРАТУРЫ

1. Sollins K. RFC 1350 – The TFTP Protocol (Revision 2) [Электронный ресурс] // Internet FAQ Archives: Online Education. – Электрон. текст. дан. – URL: <http://www.faqs.org/rfcs/rfc1350.html> (дата обращения: 04.09.2011).
2. Zhigulin M., Yevtushenko N., Maag S., and Cavalli A. // 11th International Conference on Quality Software. – Madrid, 2011. – P. 141–150.
3. Apache Commons Net [Электронный ресурс] – Электрон. текст. дан. – The Apache Software Foundation, 2001–2011. – URL: <http://commons.apache.org/net> (дата обращения: 06.09.2011).
4. Advanced TFTP [Электронный ресурс] – Электрон. текст. дан. – Geeknet, 2011. – URL: <http://freecode.com/projects/atftp> (дата обращения: 06.09.2011).

Национальный исследовательский Томский государственный университет,
г. Томск, Россия
E-mail: maxzh81@gmail.com

Поступила в редакцию 20.07.12.