УДК 51.621.391

А.Н. МАРТЫНЮК, ВАСИМ АЛЬ ШАРИФ

Одесский национальный политехнический университет, Украина

АНАЛИЗ ПРОТОКОЛА Wi-Fi ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ

Рассмотрена методика моделирования, спецификации и верификации протоколов информационных вычислительных сетей. Показан способ использования автоматной модели в качестве формальной спецификации коммуникационного протокола. Приведены свойства и соответствующие им формулы, которым должен отвечать любой протокол. Представлен способ использования автоматной модели в качестве исходной модели для метода проверки моделей (model checking). Показан фрагмент реализации модели протокола передачи. Практическим результатом работы является верификация протокола передачи с проверкой общего доступа к среде беспроводных локальных сетей (IEEE 802.11).

Ключевые слова: *верификация, протокол, проверка моделей, спецификация, автомат.*

Введение

При разработке стандарта протокола (или сервиса) сетей ЭВМ стоит задача проверки его правильности, соответствия общим свойствам всех протоколов, проверки того, что спецификации протокольных объектов вместе со спецификацией сервиса низшего уровня, соответствуют спецификации сервиса уровня разрабатываемого протокола. Первая процедура известна как анализ корректности, вторая – как верификация. Для формальной спецификации протоколов исполь-зуются автоматные модели, для сервисов – модели последовательностей [1].

Ставятся задачи спецификации DCF (Distributed Coordination Function) беспроводных сетей [2] на основе модели расширенного конечного автомата (РКА), спецификации структурных и семантических свойств РКА протокола с помощью линейной временной логики (LTL), оценки эффективности применения SPIN [3] для анализа корректности LTL-верификаторов.

1. Автоматная модель протоколов

Открытость канала беспроводных сетей приводит к множеству ошибок. Немалое внимание стандарта IEEE 802.11 [2] уделено надежности процедур передачи и доступа к распределенной среде. Базовой процедурой является DCF, в основу которой положен метод доступа с предотвращением коллизий (CSMA/CA). Передача происходит с подтверждением.

Модель станции в локальной беспроводной сети представлена РКА, граф которого изображена на рис. 1. Переходам приписаны пары (входное событие, выходная реакция), с условием, при котором выполняется переход. В вершинах состояний записаны действия, которые выполняются
после перехода к этому состоянию. РКА станций выполняют переходы параллельно.

Находясь в состоянии ожидания (IDLE), РКА реагирует на запросы протокола высшего уровня передачи (evRequest_Transmit) или получения (evRequest_Receive) сообщений. При успешном получении сообщения РКА после ожидания определенного периода (WAIT_SIFS) отвечает подтверждением.

Перед передачей РКА ждет определенное на передачу время (WAIT_DIFS) и случайный промежуток времени (BACKOFF), что уменьшает вероятность коллизий.

Если при ожидании BACKOFF канал занят (medium_busy) другой станцией, РКА ожидает конца этой передачи.

Если после ожидания канал свободен, происходит передача (TRANSMITTING) со счетчиком (SSRC) неудачных попыток.

При превышении максимума попыток передача отменяется и считается неудачной. Иначе передача удачна, и автомат опять выжидает случайный промежуток времени (BACKOFF) для предотвращения коллизий с другими станциями

Важными протокольными объектами, которые влияют на моделирование и выводы относительно правильности поведения протокола, являются таймер и канал передачи. От правильности модели таймера зависит достоверность выводов о последовательности переходов параллельных РКА. Предложенные модели изображены на рис. 2.

Если после ожидания канал свободен, происходит передача (TRANSMITTING) со счетчиком (SSRC) неудачных попыток.

При превышении максимума попыток передача отменяется и считается неудачной. Иначе передача удачна, и автомат опять выжидает случайный промежуток времени (BACKOFF) для предотвращения коллизий с другими станциями.

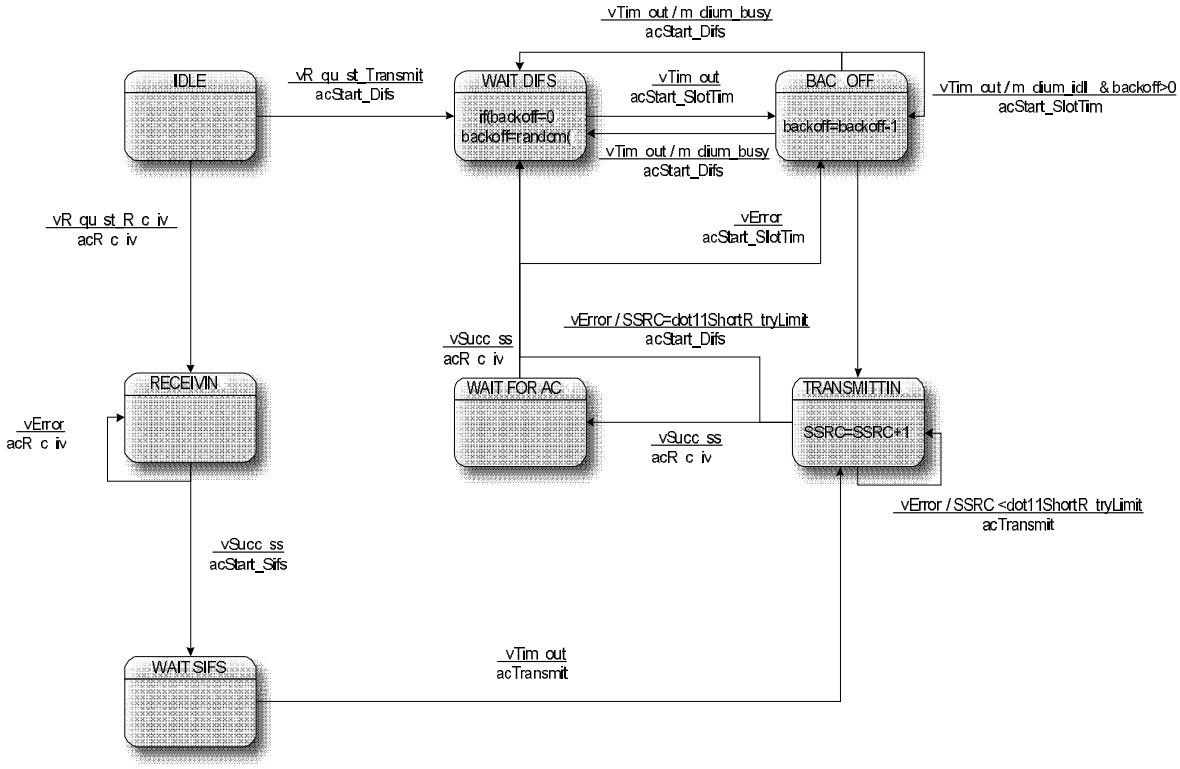
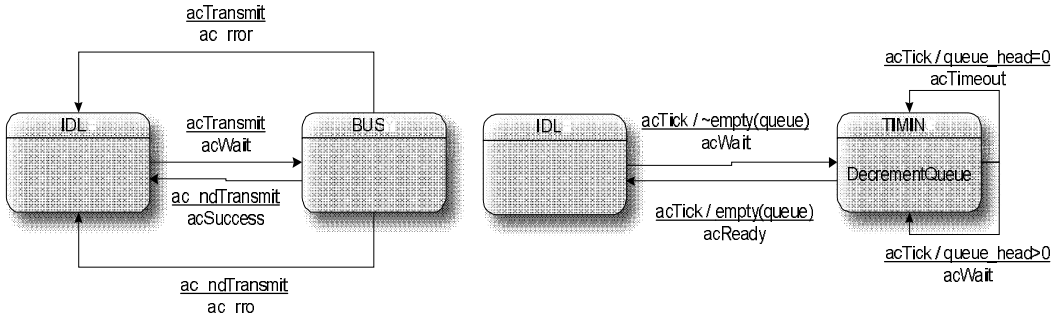


Рис. 1. Модель протокола передачи с процедурой DCF



а б

Рис. 2. Модели а – канала передачи, б – таймера

Моделью канала является РКА с двумя недетерминированными переходами с одинаковыми условиями для моделирования ошибок от случайного воздействия среды беспроводной сети.

Модель таймера является также РКА, который, кроме бесконечной очереди из сообщений счета acTick, имеет приоритетное очередь сообщений от автоматов-клиентов, ожидающих заказанные промежутки времени. В этой очереди сообщения располагаются в порядке увеличения величины промежутков времени. По каждому acTick процедура DecrementQueue отнимает от каждого промежутка значение первой, и автомат сообщает автомат, который заказывал этот промежуток, о его окончании событием acTimeout.

Таким образом, модель таймера гарантирует, что РКА-клиенты получают «таймауты» в порядке окончания промежутков времени.

2. Спецификация и верификация

РКА является удобным объектом для автоматической верификации методом проверки моделей

(model checking) [3]. Поведение модели описывается конечной системой переходов, которая называется структурой Крипке. На языке темпоральной логики возможно специфицировать свойства модели, истинность которых для данной модели затем проверяется.

Структура Крипке для системы РКА строится следующим образом. Каждому РКА Ak из набора A общей модели ставятся переменные yk, xmk, zdk, evk и stk ($0 \le k \le n$). А для всей системы РКА A в целом вводятся переменные ev, xm, zd, act, auto и evnt. Каждая из введенных переменных имеет следующий смысл:

– Yk сохраняет последнее состояние Ak;

– Xmk сохраняет последний выполненный запрос параметров управления РКА;

– Zdk сохраняет последнее выполненное исходное действие РКА Ak z ⊊ ZAk, где ZAk – множество выходных действий РКА Ak;

– Evk сохраняет последнее обработанное РКА Ak входное событие e ⊊ EAk**;**

– Stk сохраняет текущее состояние РКА Ak;

– Auto удерживает номер последнего РКА, получивший управление;

– Evnt сохраняет последнее входное событие для РКА, которому было передано управление или 0, если РКА проигнорировал последнее событие. Act сохраняет последнее элементарное событие, состоявшееся в системе РКА A. значениями act могут быть имена входных событий e ⊊ EA, входящих запросов переменных x ⊊ XA, выходных действий z ⊊ ZA, а также имена специальных действий 0 и end.

– Ev, xm и zd используются для хранения последних выполненных входных событий, входящих запросов и исходящих действий соответственно в рамках системы A в целом.

После перехода РКА в другое состояние переменная act содержит отметку этого перехода. Если РКА не может обработать входное событие, он выполняет пустое действие 0 и переходит в то самое состояние, в котором находился. При этом act равно 0. Значение end используется для идентификации безвыходного состояния всей системы РКА.

Для того, чтобы выполнить условие тотальности отношение переходов структуры Крипке, устанавливается, что из безвыходного ее состояния является лишь один переход сам в себя и при этом выполняется действие act = end.

Тогда состоянию структуры Крипке SA РКА A соответствует вектор значений переменных:

(act, auto, evnt, ev, xm, zd, ev0, xm0, zd0, y0, st0,..., evn, xmn, zdn, yn, stn).

Благодаря построенной структуре Крипке при спецификации и верификации есть возможность в качестве элементарных высказываний исполь-зовать предикаты над значениями введенных пе-ременных. В методе проверки моделей для этого используется темпоральная логика LTL [4, 5].

Свойства, которым должен отвечать любой протокол [1], и соответствующие им LTL-формулы приведены ниже.

Отсутствие статических блокировок. То есть в протоколе не существует такого положения или набора состояний, из которого не было бы переходов в другое состояние. Для системы РКА в целом это требование задается формулой

$$! <> (act == end) \text{ или } [] (act! = end),$$

а для отдельного РКА Ак

$$!<>([](auto == k -> act! = evk) \&\& []<>(auto == k)).$$

Полнота. Протокол обеспечивает реакцию на все возможные сообщения, то есть ни один РКА не проигнорирует ни одно сообщение и не выдаст пустую действие.

$$! <> (evnt == 0).$$

Следующие свойства имеют конкретный вид в зависимости от исследуемого протокола, в данной статье это требования к протоколу передачи с DCF IEEE 802.11.

Отсутствие избыточности. В спецификации протокола нет событий, не состоялись, и действий, не были выполнены. Каждая станция в сети должна завершить начатую передачу, или получения сообщения, или ждать конечный промежуток времени, или сделать конечное число попыток передачи.

<>([]<>(auto == k) && <> (act == acTransmit & & st == stTRANSMITTING-> st! = stTRANSMITTING && SSRC <dot11ShotRetryLimit))<>([]<>(auto == k))&& <>(act == acReceive && st == stRECEIVING-> st! = stRECEIVING)<> ((act == acStart_Difs || act == acStart_Sifs | | act == acStart_SlotTime | | act == acStart_WaitAck) -> evnt == evTimeout).

Ограниченность. Количество сообщений в канале ограничено. В частности, для открытого симплексного канала.

$$! <> (len (medium)> 1).$$

Завершенность. Протокол всегда достигает терминального состояния (или начального – для циклических протоколов). То есть после передачи или получения сообщения MAC-уровень станции должен вернуться в состояние ожидания последующих запросов.

$$<> (St == stIDLE-> <> (st == stIDLE)).$$

Определенная структура Крипке РКА и приведенная спецификация свойств позволяют применить для верификации метод проверки моделей. Для этого

возможно использовать программу SPIN [6], SPIN – это система верификации моделей для логики LTL «на лету» с использованием явного перечня состояний и редукции частичных порядков. Входным языком SPIN для верификации и спецификации является язык PROMELA.

Указанный формализм описания спецификации, заданной РКА, позволяет автоматизировать отображение РКА в программную реализацию, например, на языке PROMELA.

Программный комплекс автоматизированной разработки распре-деленных систем, коммуникационных протоколов и др., должен иметь транслятор описания спецификации в его реализацию, что сокращает количество ошибок при разработке и время.

3. Реализация модели

Фрагмент реализации РКА, приведенный на рис. 1, на языке PROMELA приведен ниже:

```
:: state==stBACKOFF ->
backoff--;
if
:: event==evTimeout ->
        if
        :: nempty(medium) ->
atomic{state=stWAIT_DIFS;
action=acStart_Difs;} /* medium busy */
:: empty(medium) && backoff>0 ->
atomic{state=stBACKOFF;
action=acStart_SlotTime;} /* medium idle */
        :: empty(medium) && backoff==0 &&
!end -> atomic{state=stTRANSMITTING;
action=acTransmit; mes_type=msData;} /*
medium idle and backoff finished */
:: empty(medium) && backoff==0 && end ->
atomic{state=stIDLE; action=acSuccess;} /*
postbackoff ends*/
fi
fi
:: state==stTRANSMITTING ->
        if
        :: event==evError ->SSRC++;
if
::SSRC<dot11ShortRetryLimit->
atomic{state=stTRANSMITTING;
action=acTransmit;} /* medium idle */
::SSRC==dot11ShortRetryLimit->
atomic{state=stWAIT_DIFS;action=acStart_Difs;
end=true;}
/* medium busy */
fi
:: event==evSuccess ->
atomic{state=stWAIT_FOR_ACK;
action=acReceive;}/* Transmit success. Start waiting
for ACK*/
fi
```

Проверка подлинности темпоральной свойства, заданной в виде формулы логики LTL, для модели (описанной на языке PROMELA) проводится следующим образом. Отрицание проверяемых свойств автоматически превращается в автомат Бюхи [3]. Этот автомат описывается с помощью специальной синтаксической констру-кции «never claim» языки PROMELA и задает такие пути структуры Крипке, которые не должны встречаться в модели.

Требование полноты протокола, записанное выше, на языке PROMELA имеет вид:

```
/*
 * Formula As Typed: []  (p1 && p2)
 * The Never Claim Below Corresponds
 *To The Negated Formula !([]  (p1&&p2))
 * (formalizing violations of the original)
 */
#define p1 (end_transition==true)
#define p2 (last_action!=acNoAction)
never {    /* !([]  (p1 && p2)) */
T0_init:
        if
        :: (((! ((p1))) || (! ((p2))))) -> goto accept_all
        :: (1) -> goto T0_init
        fi;
accept_all:
        skip
}
```

SPIN строит сечение РКА never claim и РКА (структуры Крипке), который выделен из программы (на языке PROMELA). Пересечение строится «на лету» без построения полной структуры Крипке. Если пересечение не пустое, выдается ложная трасса.

Заключение

В работе рассмотрена методика верификации и анализа спецификаций коммуникационных протоколов и сервисов. Спецификация задается моделью РКА, метод верификации – метод проверки моделей. Для верификации протокола передачи MAC-уровня IEEE 802.11 использован программный Верификатор SPIN.

РКА однозначно описывается на языке PROMELA, который позволяет четко и удобно ставить требования к модели и проверять их в допустимое время. В процессе верификации свойства корректности протокола подтверждены.

Литература

1. Аничкин С.А. Протоколы информационно-вычислительных сетей: Справочник. / С.А. Аничкин, С.А. Белов, А.В. Берштейн и др. – М.: Радио и связь, 1990. – 504 с.

2. LAN/MAN Committee of the IEEE Computer Society. IEEE Std 802.11™ – 2007, IEEE Standard for Information Technology. Telecommunications and information exchange between system. LANs and MANs Specific requirements. – Part 11: WLAN MAC and PHY Specifications. [Электрон. ресурс]. – Режим доступа к ресурсу: http://standards.ieee.org/getieee802/portfolio.htm.

3. Кларк Э.М. Верификация моделей программ: Model Checking. / Э.М. Кларк, О. Грамберг, Д. Пелед – М.: МЦНМО, 2002. – 416 с.

4. Monin J.-F. Understanding Formal Methods. / J.-F. Monin – Springer, 2003. – 222 p.

5. К. Ии О программных логиках – просто / К. Ии, Н.В. Шилов, Е.В. Бодин // Системная Информатика. – Наука: Новосибирск, 2002. – 316 с.

6. SPIN [Электрон. ресурс]. – Режим доступа к ресурсу: http://spinroot.com/spin/whatispin.html.

.

Поступила в редакцию 5.03.2010

Рецензент: д-р техн. наук, проф., проф. кафедры В.С. Ситников, Одесский национальный политехнический университет, Украина.

АНАЛІЗ ПРОТОКОЛУ WI-FI ОБЧИСЛЮВАЛЬНИХ МЕРЕЖ

О.М. Мартинюк, Васім Аль Шаріф

Розглянуто методику моделювання, специфікації і верифікації протоколів інформаційних мереж. Показано спосіб використання автоматної моделі як формальної специфікації комунікаційного протоколу. Приведені властивості і відповідні їм формули, яким повинен відповідати будь-який протокол. Показано спосіб використання автоматної моделі як вихідної моделі для методу перевірки моделей (model checking). Показаний фрагмент реалізації моделі протоколу передачі. Практичним результатом роботи є верифікація протоколу передачі з перевіркою загального доступу до середовища бездротових локальних мереж (IEEE 802.11).

Ключові слова: верифікація, протокол, перевірка моделей, специфікація, автомат.

ANALYZE OF NETWORK PROTOCOL Wi-Fi

O.M. Martynyuk, Waseem Al Sharif

The method of design, specification and verification of protocols of the informative computer networks is considered. The method of the using of automat model as a formal specification of communication protocol is shown. Properties and proper formulas which any protocol should correspond are resulted. The method of the using of automat model as an initial model for the method of verification of models (model checking) is presented. The fragment of implementation of the model of transmission protocol is described. A practical result is verification of transmission protocol with checking of general access to the environment of wireless local networks (IEEE 802.11).

Keywords: verification, protocol, model checking, specification, automat.

Мартынюк Александр Николаевич – канд. техн. наук, доц. кафедры компьютерных интеллектуальных систем и сетей, Одесский национальный политехнический университет, Одесса, Украина, e-mail: anmartynyuk@ukr.net.

Васим Аль Шариф – аспирант кафедры компьютерных интеллектуальных систем и сетей, Одесский национальный политехнический университет, Одесса, Украина, e-mail: waseemua@yahoo.com.