

В. А. К р и щ е н к о

**ИССЛЕДОВАНИЕ ТАЙМЕРА УДЕРЖАНИЯ
ПРИ ДИНАМИЧЕСКОЙ МАРШРУТИЗАЦИИ
НА ОСНОВЕ АЛГОРИТМА БЕЛЛМАНА–ФОРДА**

В протоколе обмена маршрутной информацией RIP существует проблема образования ложных маршрутов и маршрутных петель. Сформулирована задача нахождения интервалов значений таймеров протокола, позволяющих предотвратить образование маршрутных петель для заданной топологии сети. Предложен способ решения поставленной задачи, включающий формальное описание стандарта протокола RIP, построение по данному описанию и заданной топологии сети конечной модели и ее последующую формальную верификацию.

E-mail: kva@bmstu.ru

Ключевые слова: IP-маршрутизация, RIP, верификация протокола, таймер удержания.

Группа протоколов обмена маршрутной информацией под общим названием RIP [1, 2], достаточно широко используется в IP-сетях. Для расчета метрики маршрутов все версии протокола RIP используют распределенный вариант алгоритма Беллмана-Форда [3], который может приводить к возникновению ложных маршрутов и циклов маршрутизации.

Для уменьшения числа случаев, которые ведут к появлению таких циклов, стандарты протокола RIP содержат ряд механизмов, например таких как правило расщепленного горизонта [1]. В некоторые широко применяемые реализации протокола RIP включен также дополнительный таймер удержания [4], не входящий в текущий стандарт протокола. Существуют рекомендуемые значения таймеров протокола, призванные уменьшить вероятность возникновения ложных циклов маршрутизации в сети.

Представляет интерес вопрос, как для заданной топологии сети найти такие значения таймеров, которые исключают возможность образования ложных циклов маршрутизации. Для создания модели протокола RIP и ее применения для поиска значений таймера удержания, предотвращающих циклы, если таковые существуют в исследуемой топологии сети необходима формализация протокола. В работе [5] дано формальное доказательство корректности протокола RIP и описана его модель на языке Promela [6], но рассмотрен лишь случай отсутствия ошибок в сети, а предлагаемая модель не включала в себя таймеры протокола.

На основе формальной модели протокола обмена маршрутной информацией IPv2 создана модель протокола на языке верификации

конечных моделей Promela. С помощью разработанной модели рассматривается использование таймера удержания для минимальной топологии, подверженной образованию маршрутных петель в случае применения правила расщепленного горизонта.

Формальное описание протокола RIP. На основании стандартов [1, 2] и описания работы таймера удержания [4] можно представить более формально подмножество протокола RIP с таймером удержания. Следует отметить, что протоколы RIP2 [1] и RIPng [2] различаются главным образом форматами передаваемых сообщений и используемым сетевым протоколом, поэтому дальнейшее описание применимо к ним обоим.

В протоколе RIP маршрутизатор не хранит никакой информации о сетевой топологии, кроме таблицы маршрутизации и сведений о непосредственно подключенных к нему сетях. Пусть $L = \{0, 1, \dots, \mu\}$ — множество всех возможных значений метрик маршрутов. Протокол RIP использует метрику, равную числу маршрутизаторов до сети назначения. Значение метрики $\mu = 16$ выбрано в качестве «бесконечности» и означает недоступность сети.

Введем обозначения: R — множество всех маршрутизаторов использующей RIP системы; N — множество всех сетей системы; $E_r \subset N$ — множество сетей, подключенных к маршрутизатору r ; $A_r^k \subseteq N$ — множество сетей, известных маршрутизатору r в момент времени t_k .

Таблица маршрутизации маршрутизатора $r \in R$ в момент времени t_k является отображением:

$$\alpha_r^k : A_r^k \rightarrow R \times L \times N.$$

В таблице маршрутизации запись о маршруте является упорядоченной тройкой $\alpha_r^k(n) = \langle h, l, s \rangle$:

$n \in A_r^k$ сеть назначения данного маршрута;

$h \in R$ следующий маршрутизатор в маршруте;

$l \in L$ рассчитанная метрика маршрута;

$s \in E_r$ — сеть, из которой была получена информация о данном маршруте, если маршрутизатор не подключен к сети n непосредственно ($n \notin E_r$).

Метрика маршрута до сети не изменяется: $\forall k : \alpha_r^k(n) = \langle r, 1, n \rangle$, если маршрутизатор подключен к сети непосредственно ($n \in E_r$).

Для краткости условимся в дальнейшем опускать индекс k , когда он очевиден или неважен. В начальный момент времени маршрутизатор располагает информацией только о подключенных к нему сетях ($A_r^0 = E_r$), и начальная таблица маршрутизации содержит информацию только о них:

$$\forall n \in E_r : \alpha_r^0(n) = \langle r, 1, n \rangle.$$

В стандарте [1] заданы три операции с таблицами маршрутизации:

1) добавление в таблицу маршрута до сети n

$$A_r^k = A_r^{k-1} \cup \{n\}, \quad \alpha_r^k(n) = \langle h, l, s \rangle, \quad l < \mu;$$

2) пометка сети n как временно недостижимой

$$\alpha_r^k(n) = \langle h, \mu, s \rangle;$$

3) удаление информации о маршруте до сети n из таблицы

$$A_r^k = A_r^{k-1} \setminus \{n\}.$$

Пусть $C_{r,d} = E_r \cap E_d$ — множество таких сетей, к которым непосредственно подключены оба маршрутизатора r и d . Пусть $H_r^v = \langle d \in R | v \in C_{r,d} \rangle$ — множество всех маршрутизаторов, доступных от маршрутизатора r через сеть v с помощью только непосредственной маршрутизации. Тогда маршрутизатор $r \in R$ регулярно рассылает сообщения с маршрутной информацией

$$M_r^v \subseteq A_r \times \{r\} \times L$$

через сеть $v \in C_{r,d}$ каждому своему соседу $d \in H_r^v$. Для рассылки сообщений используется широковещательная рассылка.

Если применяется правило «расщепленный горизонт с недопустимым обратным маршрутом», то маршрутизатор проверяет, не совпадает ли сеть v , в которую направляется сообщение M_r^v , с сетью, от которой получена информация о рассылаемом маршруте. При их совпадении, то в качестве метрики маршрута в сообщении указывается «бесконечность» как средство борьбы с маршрутными циклами. Поэтому сообщение с маршрутной информацией M_r^v формируется по следующим правилам:

$$\forall n \in A_r : (\alpha_r(n) = \langle h, l, s \rangle \wedge s \neq v) \Rightarrow \langle n, r, l \rangle \in M_r^v;$$

$$\forall n \in A_r : (\alpha_r(n) = \langle h, l, s \rangle \wedge s = v) \Rightarrow \langle n, r, \mu \rangle \in M_r^v.$$

При получении сообщений, содержащих маршрутную информацию, маршрутизатор выполняет некоторые действия со своей таблицей маршрутизации. Выделим следующие события, связанные с получением информации о маршруте:

событие σ_{new} происходит, когда получена информация о новом маршруте;

событие σ_{same} — когда получена информация о том же маршруте от того же маршрутизатора;

событие σ_{shorter} — когда получена информация о более коротком пути до уже известной сети;

событие σ_{inf} — когда получено сообщение, что уже известный путь стал недоступен.

Пусть $M_{h,r}^s$ — сообщение о маршрутах, полученное маршрутизатором r от маршрутизатора h через сеть s . В соответствии со спецификацией протокола

$$\begin{aligned} (\langle n, h, l \rangle \in M_{h,r}^s \wedge n \notin A_r \wedge l < \mu - 1) &\Rightarrow \sigma_{\text{new}}; \\ (\langle n, h, l \rangle \in M_{h,r}^s \wedge \alpha_r(n) = \langle h, l_{\text{old}}, s_{\text{old}} \rangle \wedge l < \mu - 1) &\Rightarrow \sigma_{\text{same}}; \\ (\langle n, h, l \rangle \in M_{h,r}^s \wedge \alpha_r(n) = \langle h_{\text{old}}, l_{\text{old}}, s_{\text{old}} \rangle \wedge l < l_{\text{old}}) &\Rightarrow \sigma_{\text{shorter}}; \\ (\langle n, h, l \rangle \in M_{h,r}^s \wedge \alpha_r(n) = \langle h, l_{\text{old}}, s_{\text{old}} \rangle \wedge l \geq \mu - 1) &\Rightarrow \sigma_{\text{inf}}. \end{aligned}$$

Помимо таймера $T_{\text{send}} = 30$ с, управляющего периодичностью рассылки сообщений, каждый маршрутизатор должен использовать ряд таймеров, связанных с каждым маршрутом. Каждый элемент маршрутной таблицы с метрикой $1 < l < \mu$ связан с запущенным таймером истечения времени маршрута, по умолчанию имеющим значение $T_{\text{route}} = 180$ с.

Маршрутизатор сбрасывает этот таймер, когда случается одно из следующих событий: σ_{new} , σ_{same} , или σ_{shorter} . Каждый маршрут к недостижимой сети с метрикой $l = \mu$ связан с запущенным таймером уборки мусора, имеющим по умолчанию значение $T_{\text{flush}} = 120$ с. В случае применения таймера удержания каждая запись о недостижимой сети связана также с таймером удержания, равным по умолчанию $T_{\text{hold}} = 180$ с. При использовании таймера удержания должно выполняться условие $T_{\text{hold}} \leq T_{\text{flush}}$, поэтому при использовании таймера удержания рекомендуется выбрать значение $T_{\text{flush}} = 240$ с.

Пусть β_r^k — отображение между маршрутами, известными маршрутизатору r , и текущими значениями их таймеров в момент времени t_k :

$$\beta_r^k : A_r^k \setminus E_r \rightarrow [0, T_{\text{route}}] \times [0, T_{\text{flush}}] \times [0, T_{\text{hold}}].$$

Свяжем следующие события с моментами срабатывания таймеров маршрута:

событие σ_{expired} происходит в момент срабатывания таймера истечения времени маршрута;

событие σ_{flush} — в момент срабатывания таймера уборки мусора;

событие σ_{hold} — в момент срабатывания таймера удержания.

Рассмотрим возможные состояния маршрута до сети n от маршрутизатора r . Жизненный цикл маршрута можно описать в качестве конечного автомата с выходом (рис. 1), который определяется кортежем

$$F_n = \langle \Sigma, \Gamma, S, s_0, \delta, \omega \rangle,$$

где $\Sigma = \{\sigma_{\text{new}}, \sigma_{\text{same}}, \sigma_{\text{shorter}}, \sigma_{\text{inf}}, \sigma_{\text{expired}}, \sigma_{\text{hold}}, \sigma_{\text{flush}}\}$ — входной алфавит автомата; $\Gamma = \{\gamma_{\text{valid}}, \gamma_{\text{hold}}, \gamma_{\text{remove}}, \gamma_{\text{none}}\}$ — выходной алфавит автомата; $S = \{s_{\text{valid}}, s_{\text{hold}}, s_{\text{invalid}}, s_{\text{unknown}}\}$ — множество состояний

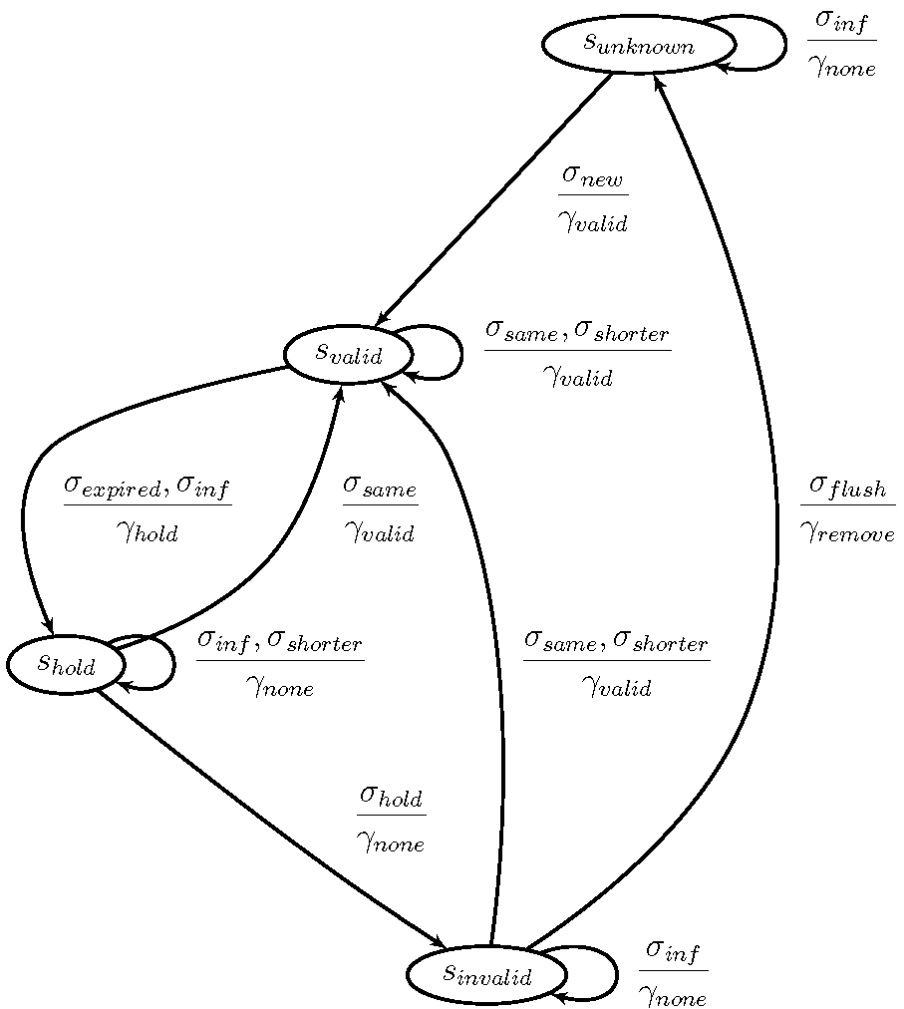


Рис. 1. Конечный автомат состояний маршрута в протоколе RIP

маршрута; начальное состояние $s_0 = s_{valid}$, если $n \in E_r$ (маршрутизатор подключен к сети непосредственно); начальное состояние $s_0 = s_{unknown}$, если $n \notin E_r$; $\delta : S \times \Sigma \rightarrow S$ — функция переходов (см. рис. 1); $\omega : S \times \Sigma \rightarrow \Gamma$ — функция выхода (см. рис. 1).

Символы выходного алфавита автомата имеют следующий смысл:

γ_{valid} — в таблицу маршрутизации добавлена информация о достижимой сети;

γ_{hold} — сеть отмечена в маршрутной таблице как недоступная;

γ_{remove} — маршрут удален из таблицы в ходе уборки мусора;

γ_{none} — маршрутная таблица не изменяется.

Тогда выходной алфавит можно рассматривать как множество предикатов, причем верны утверждения:

$$\gamma_{valid} \Rightarrow (\alpha_r^k(n) = \langle h, l + 1, s \rangle \wedge \beta_r^k(n) = \langle T_{expire}, 0, 0 \rangle);$$

$$\gamma_{\text{hold}} \Rightarrow (\alpha_r^k(n) = \langle h, \mu, s \rangle \wedge \beta_r^k(n) = \langle 0, T_{\text{flush}}, T_{\text{hold}} \rangle);$$

$$\gamma_{\text{remove}} \Rightarrow (A_r^k = A_r^{k-1} \setminus \{n\}).$$

Кроме того, при событии γ_{valid} сбрасывается таймер устаревания маршрута, а при событии γ_{hold} запускаются таймеры удержания и таймер сборки мусора.

Проблема образования маршрутных петель. На рис. 2 показана наименьшая вычислительная сеть, в которой образования циклов в случае применения стандартного протокола RIP2 даже при использовании правила расщепленного горизонта.

В работах [3, 5] доказано, что при отсутствии ошибок в сети каждый маршрутизатор RIP создаст полную верную таблицу маршрутизации в течение конечного отрезка времени. Когда таблица сформируется в системе будут постоянно рассылаться сообщения о маршрутах до сети v (см. рис. 2); маршруты до прочих сетей не показаны для упрощения примера. Допустим, что затем происходят следующие события.

1. Сеть w перестает нормально функционировать, после чего срабатывает таймер истечения маршрута (событие σ_{expired}) до сети v у маршрутизатора b , тогда $\alpha_b(v) = \langle a, \mu, w \rangle$.

2. Маршрутизатор b отправляет сообщение $\langle v, b, \mu \rangle \in M_{b,c}^x$, $\langle v, b, \mu \rangle \in M_{b,d}^y$.

3. Маршрутизатор c отправляет сообщение $\langle v, c, 3 \rangle \in M_{c,d}^z$.

4. Маршрутизатор d принимает сообщение $\langle v, b, \mu \rangle \in M_{b,d}^y$, в результате $\alpha_d(v) = \langle b, \mu, y \rangle$ (рис. 3).

5. Маршрутизатор d принимает сообщение $\langle v, c, 3 \rangle \in M_{c,d}^z$, в результате $\alpha_d(v) = \langle c, 4, z \rangle$. Маршрутизатор c принимает сообщение $\langle v, b, \mu \rangle \in M_{b,c}^x$.

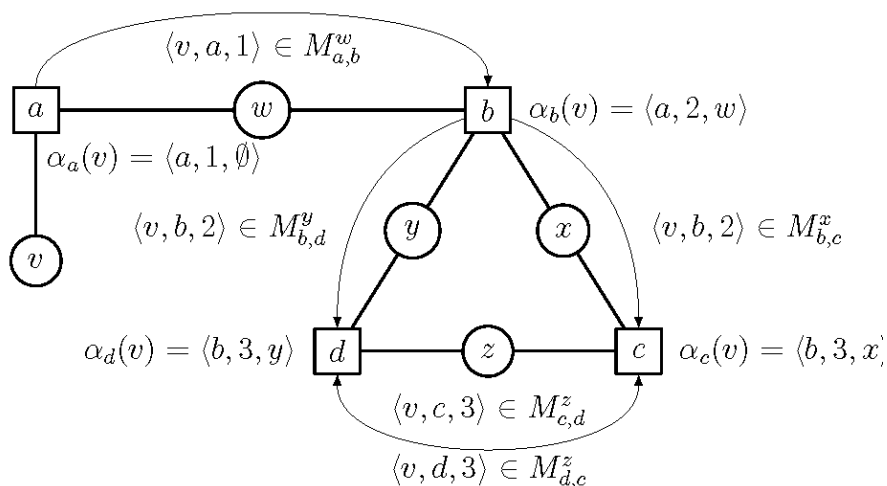


Рис. 2. Система, в которой могут возникать маршрутные петли

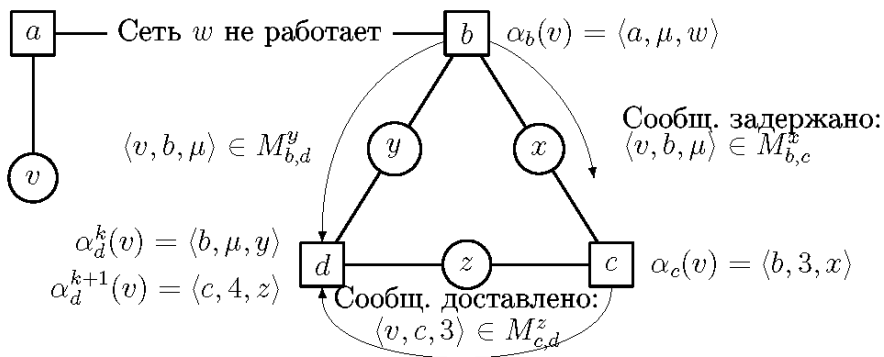


Рис. 3. Первая стадия образования маршрутной петли

6. Маршрутизатор d отправляет сообщение $\langle v, d, 4 \rangle \in M_{d,b}^y$.

7. Маршрутизатор b принимает сообщение $\langle v, d, 4 \rangle \in M_{d,b}^y$, в результате $\alpha_b(v) = \langle d, 5, y \rangle$ (рис. 4). Маршрутизатор b начинает сообщать об этом маршруте маршрутизатору c . Таким образом, сформирована ложная маршрутная петля (см. рис. 4).

Кроме правила расщепленного горизонта, в протоколе RIP существуют два способа борьбы с маршрутными петлями — инициированные обновления [1], когда маршрутизатор рассылает сообщения об измененном маршруте в течение нескольких секунд, и таймер удержания [4], задающий момент времени, в течение которого маршрутизатор игнорирует сообщения о доступных маршрутах сети, недавно ставшей недоступной.

Предположим теперь, что в рассмотренной системе используется протокол RIP с включенными инициированными обновлениями. Если инициированное сообщение от маршрутизатора b к маршрутизатору c потеряно, то ситуация будет развиваться аналогично тому, как описано выше начинается с шага 4. Кроме того, даже без потери сообщения может иметь место случай, когда волна инициированных обновлений пересечется с регулярными сообщениями: например, маршрутизатор c

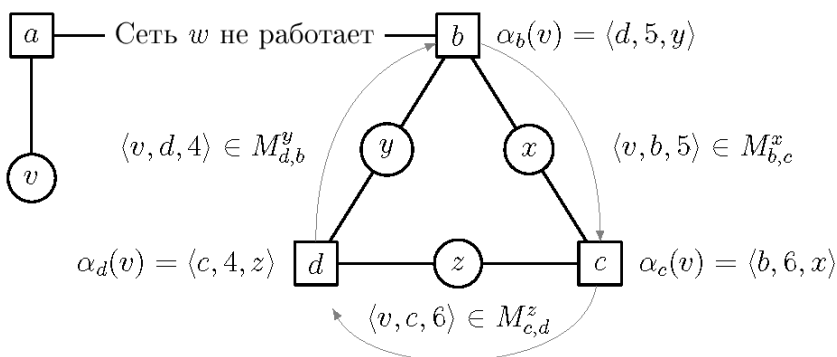


Рис. 4. Вторая стадия образования маршрутной петли

успеет отправить регулярное сообщение маршрутизатору d в момент между получением маршрутизатором d иницированного сообщения от b и до получения c такого же сообщения.

Из изложенного выше следует, что использование иницированных сообщений не приведет к устранению возможности возникновения циклов маршрутизации, а только снизит вероятность их образования. Использование же таймеров удержания при некоторых их значениях, возможно, могло бы привести к предотвращению зарождения циклов.

Модель взаимодействия RIP-маршрутизаторов. Автономная система, в которой применяется протокол обмена маршрутной информацией RIP с точки зрения задачи нахождения безопасных значений таймеров представляет собой набор асинхронно взаимодействующих маршрутизаторов.

Протокол RIP использует протокол ненадежной передачи без установки соединения UDP. Искажение передаваемых данных сводится к их потере, поскольку в UDP производится проверка контрольной суммы, но отсутствует повтор передачи. Объем передаваемых данных весьма мал: при стандартном MTU сети маршрутизатор RIP посылает всю свою таблицу маршрутизации в единственном IP-пакете, если в таблице не более 25 записей [1], поэтому сообщение, содержащее информацию о маршрутах, либо корректно принимается целиком, либо не доходит до процесса службы RIP вообще, пока $|N| < 25$. В связи с этим, модель передачи данных должна имитировать следующие виды возможного взаимодействия служб маршрутизации:

- успешную доставку целого сообщения RIP через протокол UDP;
- случайную потерю сообщений RIP, передаваемых между двумя маршрутизаторами.

При отсутствии иницированных обновлений маршрутизатор RIP не проявляет никакой активности между срабатываниями таймера отправки. Стандартное значение таймера отправки $T_{\text{send}} = 30$ достаточно велико, чтобы за это время сообщение прошло путь между соседними маршрутизаторами. В силу этого достаточно рассматривать два случая успешной доставки сообщения (рис. 5):

- между отправкой сообщения и его приемом на маршрутизаторе-получателе срабатывает таймер отправки и происходит рассылка сообщений;
- между отправкой сообщения и его приемом таймер отправки маршрутизатора-получателя не успевает сработать.

Сообщения хранятся в очереди, связанной с процессом маршрутизатора, и будут извлечены процессом-получателем либо перед ближайшей рассылкой, либо перед последующей.

Моделирование ошибок в сети передачи данных происходит следующим образом. Модель каждой сети $n \in N_{\text{error}} \subseteq N$ находится в

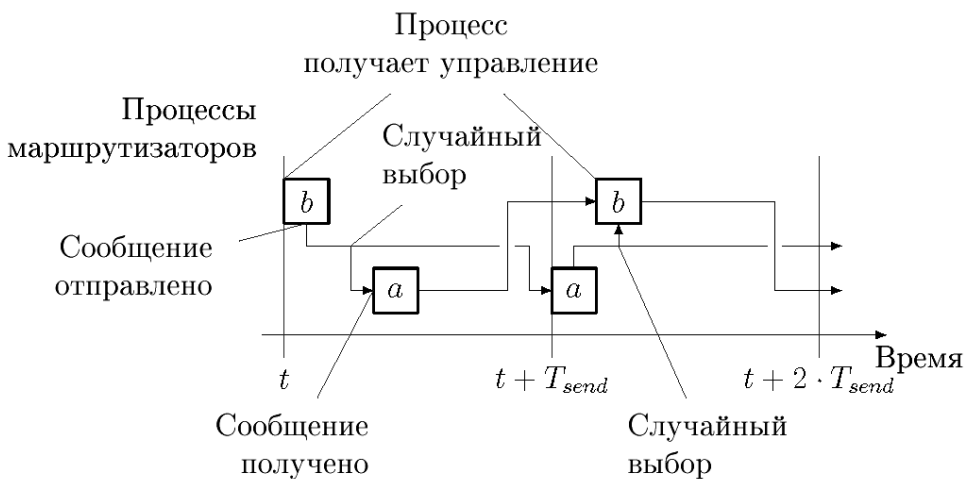


Рис. 5. Отправка и получение сообщений протокола RIP

одном из двух состояний: данные либо передаются по сети корректно, либо безвозвратно теряются. Сети из множества $N \setminus N_{\text{error}}$ работают всегда корректно: таковыми, например, с целью уменьшения состояний системы могут считаться сети, подключенные только к одному маршрутизатору. Каждая сеть $n \in N_{\text{error}}$ случайно изменяет или сохраняет свое состояние каждый раз, когда какой-либо маршрутизатор посылает сообщения, содержащие известные ему маршруты, своим соседям или после отправки сообщения каждому соседу.

Реализация модели системы, использующей протокол RIP. Сеть из маршрутизаторов, использующих протокол RIP, можно рассматривать как систему с конечным числом состояний, состоящую из независимых процессов. Поскольку в протоколе RIP отсутствует какая-либо синхронизация между маршрутизаторами, то передачу сообщений между ними можно моделировать с помощью асинхронных каналов. Такая система является конечной, поскольку любая использующая протокол RIP сеть не может иметь «диаметр», превышающий 15 маршрутизаторов. Таким образом, верифицируемая система является распределенной системой с ограниченным числом процессов.

Метод верификации конечных моделей путем полного перебора состояний [7] может быть применен для решения поставленной задачи при зафиксированных значениях таймеров T_{route} , T_{hold} и T_{flush} . Для рассмотренного примера очевидным нарушением корректности системы будет $\alpha_b(v) = \langle c, l, x \rangle \vee \alpha_b(v) = \langle d, l, y \rangle$.

Интервалы всех таймеров протокола RIP выбираются с периодом, равным значению таймера рассылки сообщения $T_{\text{send}} = 30$ с, причем значения интервалов таймеров не превышают нескольких минут. Поэтому, множество возможных значений любого таймера протокола содержит лишь несколько элементов, и при верификации можно

воспользоваться перебором всех возможных значений таймеров T_{route} , T_{hold} и T_{flush} .

Для описания верифицируемой модели выбран язык Promela [6] как достаточно распространенный язык описания конечных моделей, состоящих из конкурирующих процессов. Рассмотренный выше упрощенный вариант формализации протокола выбран в качестве исходного. В частности, в модели хранятся и передаются маршруты только до единственной сети, поскольку при полном переборе состояний системы значительно эффективнее оценивать маршруты до каждой сети в отдельных экспериментах в силу их независимости.

Поскольку в протоколе RIP используется групповая рассылка и простой транспортный протокол UDP, то в случае возникновения проблем в сети рассылающая сторона не предпринимает никаких специальных действий. При поиске условий для полного устранения возникновения маршрутных петель в качестве модели ошибок можно рассмотреть следующую:

- сеть может быть исправна (сообщение передается корректно) или неисправна (сообщение теряется);
- сеть изменяет свое состояние случайным образом перед передачей по ней сообщения протокола RIP.

При необходимости можно ограничить максимальное число смены состояний для каждой сети.

Результаты экспериментов. Созданная модель была использована для проведения экспериментов с рассмотренной сетью, состоящей из четырех маршрутизаторов (см. рис. 2). Эта сеть известна тем, что в ней существует вероятность возникновения циклов маршрутизации в случае использования протокола RIP без таймера удержания.

В ходе вычислительных экспериментов проверялись все комбинации из следующих возможных значений таймеров с учетом ограничения $T_{hold} \leq T_{flush}$: $\langle T_{route}, T_{flush}, T_{hold} \rangle \in \{60, 90, 120, 150\} \times \{120, 150, 180, 210, 240\} \times \{120, 150, 180, 210, 240\}$. Для исследуемой сети были рассмотрены две модели ошибок: в первом случае сети x , y , z изменяли свое состояние сколь угодно раз, во втором — не более двух раз.

В таблице приведены только те комбинации значений таймеров, которые успешно прошли проверку и которые имеют минимальное значение T_{hold} для данных T_{flush} и T_{route} . В обоих вариантах модели ошибок сети были получены одинаковые допустимые значения таймеров. Как следует из таблицы, для рассматриваемой сети в случае совпадающих значений таймеров у всех маршрутизаторов для предотвращения образования циклов должно выполняться условие $T_{route} \leq \frac{T_{hold}}{2}$. В результате применения созданного метода могут быть найдены условия,

Результаты проверки значений таймеров

T_{route} , с	T_{flush} с	T_{route} с
60	120	120
60	150	120
60	180	120
60	210	120
60	240	120
90	180	180
90	210	180
90	240	180
120	240	240

при которых исключается возможность образования ложных циклов маршрутизации в заданной топологии сети.

СПИСОК ЛИТЕРАТУРЫ

1. M a l k i n G. RFC 2453: RIP version 2. Internet Standard. 1998.
2. M a l k i n G., M i n n e a r R. RFC 2080: RIPng for IPv6. Internet Standard. 1997.
3. B e l l m a n R. On a Routing Problem // Quarterly of Applied Mathematics – 1958. – Vol. 16, No. 1. P. 87–90.
4. B r u n o A., K i m J. CCDA Exam Certification Guide, 2nd Edition – Cisco Press, 2003. – 696 p.
5. B h a r g a v a n K., G u n t e r C. A., O b r a d o v i c h D. Routing Information Protocol in HOL/SPIN // Proc. of the 13th Int. Conf. on Theorem Proving in Higher Order Logics (TPHOLs '00). 2000. – P. 53–72.
6. H o l z m a n n G. The SPIN Model Checker: Primer and Reference Manual. – Addison-Wesley, 2003. – 608 p.
7. К л а р к Э. М., Г р а м б е р г О., П е л е д Д. Верификация моделей программ. – М.: МЦНМО, 2002. – 416 с.

Статья поступила в редакцию 10.05.2012