Опубликовано: Зв'язок. - 2006, Т. 64, №4. - С. 49-58

Верификация протокола TCP в процессе последовательной композиции модели Петри

Д.А.Зайцев

Одесская национальная академия связи им. А.С.Попова
Ул. Кузнечная 1, Одесса 65038, Украина
http://www.geocities.com/zsoftua

Аннотация

Доказана корректность процедур установления соединения и разъединения протокола TCP. Модель, построенная по стандартным спецификациям протокола, представлена сетью Петри. Особенностью модели является раздельное представление флагов стандартного заголовка сообщения протокола TCP, используемых в процедурах установления соединения и разъединения. Для верификации протокола применены инварианты сетей Петри. Вычисление инвариантов выполнено в процессе последовательной композиции модели из функциональных подсетей, обеспечивающей существенное ускорение вычислений.

Ключевые слова: *TCP, Сеть Петри, Инвариант, Функциональная подсеть, Композиция.*

1. Введение

Трудно переоценить значимость протокола TCP [1,2] для обеспечения коммуникации информации в современном мире. TCP является основным транспортным протоколом Internet. Именно посредством TCP ежедневно передаётся более двухсот петабит информации, как общего пользования, так и корпоративной. В семействе протоколов TCP/IP на долю TCP приходится около 80% всего информационного потока и лишь оставшиеся 20% всего объёма информации доставляется посредством протокола UDP. Таким образом, формальное доказательство корректности протокола TCP имеет ключевое значение для обоснования надёжности функционирования современных глобальных сетей. В условиях вирусных атак необходима уверенность в том, что исходные спецификации протоколов не содержат скрытых дефектов. Это обусловлено тем, что механизм внедрения большинства интеллектуальных вирусов основан на использовании дефектов либо в программном обеспечении, реализующем определённые протоколы, либо в самих спецификациях протоколов.

Верификация протоколов является традиционной областью применения сетей Петри [3]. Первые работы представляющие методы верификации протоколов с помощью сетей Петри появились в начале восьмидесятых годов прошлого столетия [4,5]. Действительно, большинство сетевых протоколов предполагают асинхронный параллельный характер взаимодействия систем, что затрудняет их описания с помощью последовательных моделей, таких, например, как блок-схема. При исследовании протоколов решают две основные задачи: доказательство корректности протокола и оценку его эффективности. Первую из задач называют ещё верификацией протокола. Она действительно является приоритетной, поскольку наличие дефектов в исходных спецификациях представляет собой один из наиболее дорогостоящих типов ошибок. Поскольку, если некорректный протокол будет реализован программно либо аппаратно, то затраты на исправление ошибок могут быть весьма значительными.

В процессе верификации, как правило, устанавливается возможность неограниченно повторять процесс взаимодействия систем, используя при этом элементы ограниченной ёмкости

[6,7]. В негативной формулировке эти условия можно представить, как отсутствие взаимных блокировок (тупиков), а также отсутствие последовательностей действий, приводящих к переполнению накопителей. Детализированные модели реальных телекоммуникационных протоколов, представленные сетями Петри и построенные по исходным спецификациям, насчитывают, как правило, сотни элементов. Такая большая размерность создаёт значительные трудности в применении формальных методов исследования свойств модели, позволяющих доказать корректность протокола.

Одними из наиболее распространённых методов исследования свойств сетей Петри является метод инвариантов [3]. Реализация этого метода сводится к нахождению целых неотрицательных решений систем линейных диофантовых уравнений. Заметим, что поиск целых неотрицательных решений линейной системы представляет собой специфическую задачу, для решения которой предложены специальные методы [8,9]. К сожалению, сложность этих методов асимптотически экспоненциальна, что делает практически невозможным поиск инвариантов для сетей, насчитывающих более сотни элементов.

В [10] представлен полиномиальный алгоритм декомпозиции заданной сети Петри на функциональные подсети. В [11] инварианты функциональных подсетей использованы для построения инвариантов исходной сети; в [12,13] решена задача последовательной композиции функциональных подсетей. Показано, что полученные ускорения вычислений является экспоненциальным по отношению к количеству вершин сети, что позволяет выполнить анализ моделей, для которых он ранее расценивался как практически неосуществимый. Применение методов вычисления инвариантов в процессе одновременной и последовательной композиции сети Петри из её функциональных подсетей изучено на примере верификации протокола BGP в [14,15]. Однако, в [14,15] использована упрощённая модель протокола. При построении модели не рассматривалась структура передаваемых сообщений; каждое сообщение моделировалось единственной фишкой сети Петри.

Целью настоящей работы является построение детализированной модели Петри протокола TCP, учитывающей формат заголовка сообщения, для формального доказательства корректности фаз установления соединения и разъединения. Спецификации протокола кроме указанных двух фаз определяют ещё основную фазу обмена информацией. Её исследование требует применения расширенных сетей Петри [16] и выходит за рамки настоящей работы.

2. Описание протокола TCP

Стандартная спецификация протокола TCP была представлена в 1981 году в RFC 793 [1]. Этот документ явился результатом продолжительных обсуждений, отражённых, например, в RFC с номерами 44, 55, 761. В процессе эксплуатации в протокол были внесены изменения, касающиеся таких вопросов как медленный старт RFC 1122, быстрое восстановление RFC 2001, повторная передача RFC 2988. Совершенствование стандарта не прекращается и в настоящее время, о чём свидетельствуют, например, документы RFC 3360, 3481, 3562, в которых предлагаются средства надежного взаимодействия при сбросе связи, обмен информацией по беспроводным линиям, алгоритмы обмена ключами для обеспечения защиты информации.

Напомним, что протокол TCP предназначен для транспортировки потоков данных в двух направлениях между парой приложений, каждое из которых запущено на компьютере, подключенном к сети. В отличие от протокола UDP, протокол TCP предполагает установление соединения между взаимодействующими приложениями. Для однозначной идентификации приложения используется концепция гнезда (socket). Гнездо представляет собой пару, состоящую из IP-адреса и номера порта. IP-адрес является элементом протокола IP,

расположенного на сетевом уровне иерархии ISO, и однозначно определяет хост сети. Порт представляет собой логический номер, идентифицирующий процесс внутри хоста.

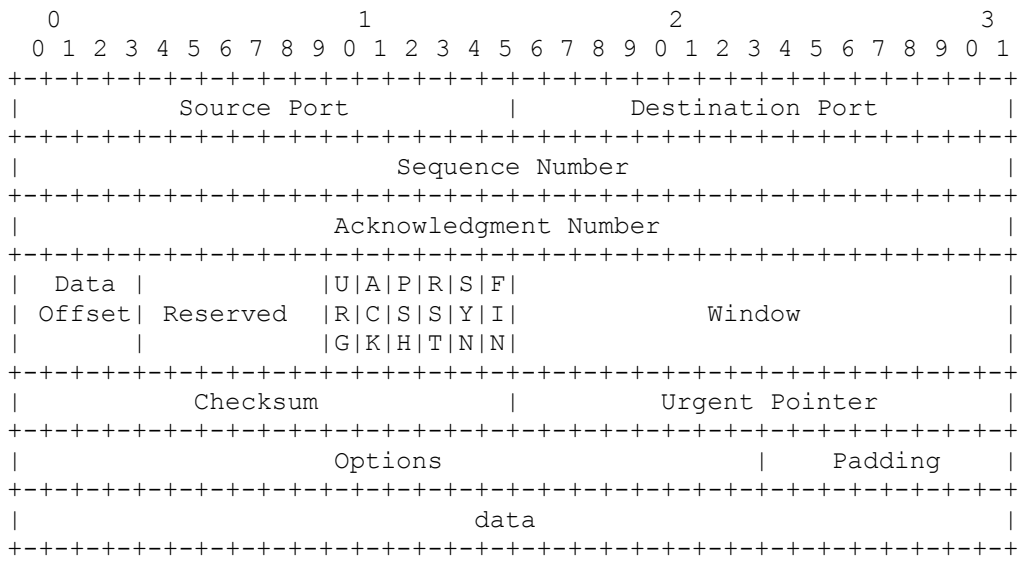


Рис. 1. Формат заголовка TCP

Протокол TCP имеет три фазы: установление соединения, передача данных и разъединение. Формат заголовка протокола TCP, предусмотренный стандартом, представлен на Рис. 1. Отметим, что фазы установления соединения и разъединения используют битовые флаги заголовка SYN, FIN, ACK. Остальные поля заголовка предназначены для управления обменом информацией и позволяют контролировать скорость передачи данных, их корректность, а также обеспечивать повторную передачу искажённой информации. Флаг SYN применяют для синхронизации номеров последовательностей (Sequence Number) передаваемых данных при установлении соединения. Флаг FIN инициирует завершение связи. Флаг ACK используют для подтверждения; он может быть указан совместно с другими флагами.

Диаграмма состояний протокола TCP, предусмотренная стандартом, представлена на Рис. 2. Отметим, что состояния изменяются с одной стороны под влиянием команд, поступающих от протоколов прикладного уровня, таких как OPEN, CLOSE, SEND, RECEIVE, а с другой – под воздействием полей заголовков пакетов, доставляемых протоколом сетевого уровня. Предусмотрено два типа открытия связи с помощью команды OPEN: активная (active) – когда выполняется попытка установления соединения с указанным гнездом и пассивная (passive) – когда осуществляется прослушивание указанного гнезда в целях ожидания входящих запросов на соединение. Активные команды OPEN применяют, как правило, в клиентском программном обеспечении, пассивные – в серверном. Текущее состояние связи полностью контролируется управляющим блоком TCB.

Сокращения snd и rcvd использованы для представления посылки и получения пакета с указанным признаком соответственно. Основными являются состояния автономной работы (CLOSED), прослушивания (LISTEN), установленного соединения (ESTAB). Отметим, что обмен информацией выполняется в состоянии ESTAB. Процедура установления соединения протокола TCP известна также под названием тройного рукопожатия (three-way handshake) так как требует обмена тремя начальными пакетами. Разъединение также выполняется в три этапа.

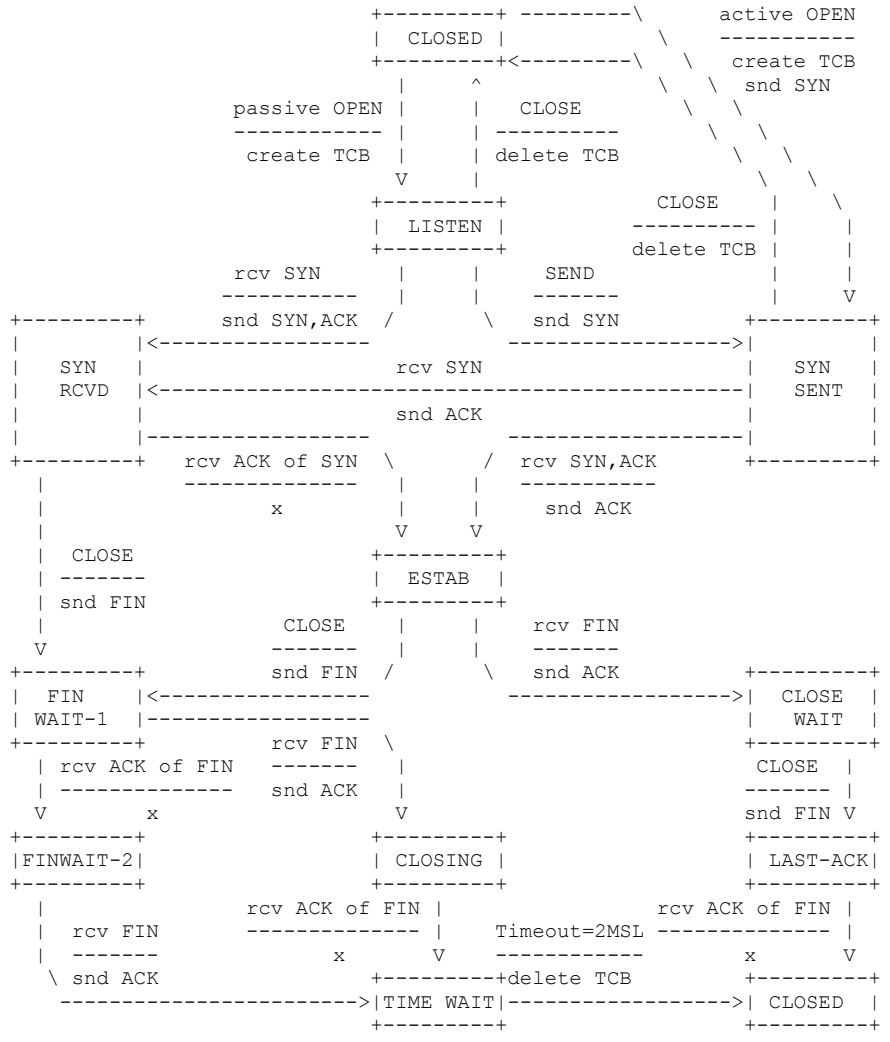


Рис. 2. Диаграмма состояний протокола TCP

Таким образом, даже если не рассматривать фазу обмена информацией, протокол должен обеспечивать корректную обработку каждой из четырёх указанных команд и каждого из трёх флагов в произвольные моменты времени. Наличие кроме трёх основных ещё девяти дополнительных состояний делает процесс достаточно сложным и требует применения специальных формальных методов.

2. Модель Петри протокола TCP

Модель протокола TCP в форме сети Петри изображена на Рис. 3. Напомним, что *сеть Петри* [3] – это тройка $N = (P, T, F)$, где $P = \{p\}$ – конечное множество вершин, называемых позициями, $T = \{t\}$ – конечное множество вершин, называемых переходами, отношение смежности вершин $F \subseteq P \times T \bigcup T \times P$ задаёт множество дуг, соединяющих позиции и переходы. Таким образом, сеть Петри представляет собой двудольный ориентированный граф, одну долю вершин которого составляют позиции, а другую – переходы. Позиции изображают окружностями, а переходы – прямоугольниками. Как правило, граф N дополняют функцией разметки, задающей первоначальное расположение фишек в позициях. Фишки представляют

собой динамические элементы, которые перемещаются по сети в результате срабатывания переходов [3].

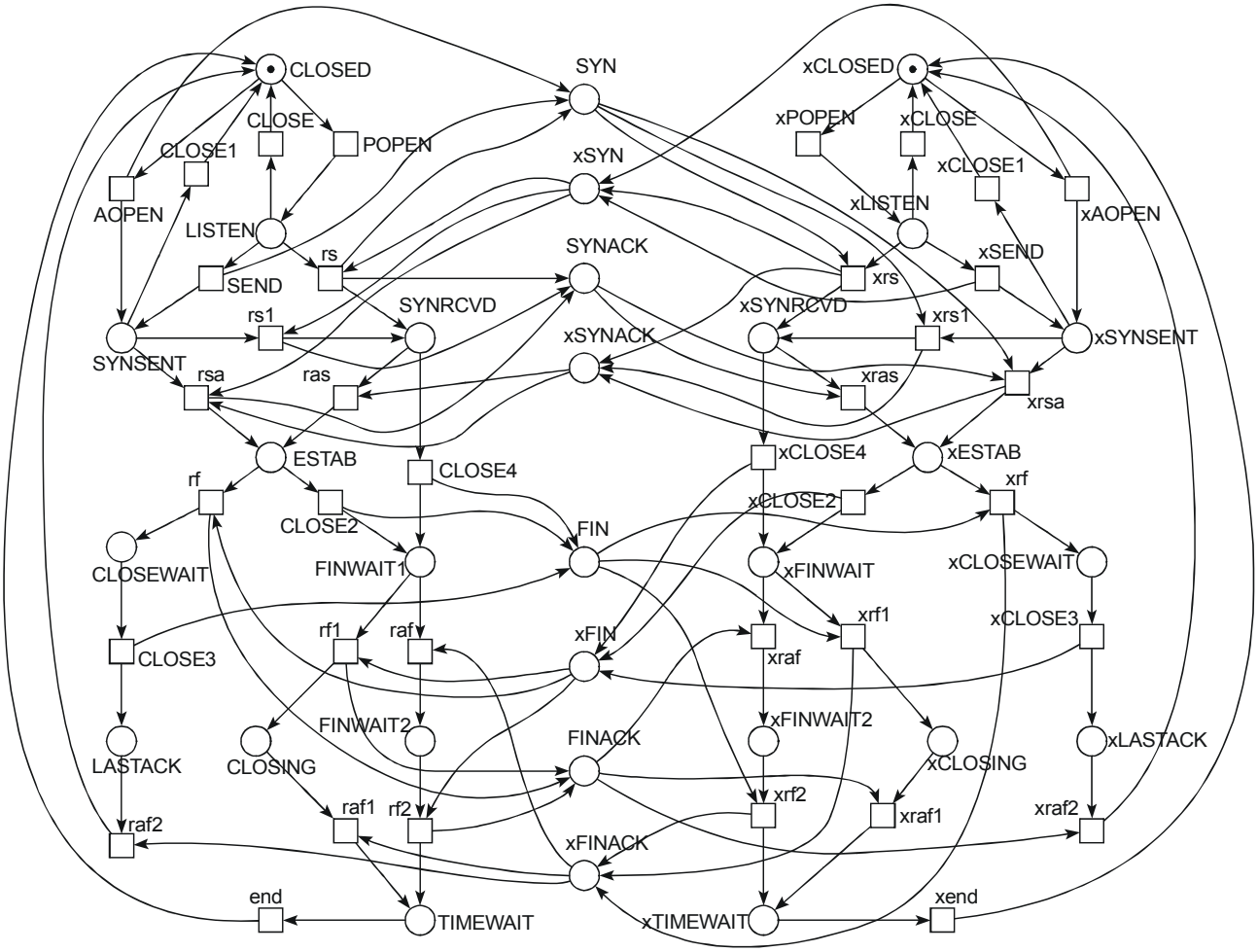


Рис. 3. Модель Петри протокола TCP

В модели, представленной на рис. 3, можно выделить три основных части: левая взаимодействующая система; правая взаимодействующая система; коммуникационная подсистема. Каждая из взаимодействующих систем в точности соответствует стандартной диаграмме состояний (рис. 2) протокола. В обозначениях правой подсистемы присутствует префикс x. Состояния диаграммы представлены одноимёнными позициями сети Петри. Дуги диаграммы состояний представлены переходами сети Петри. При этом использованы отдельные позиции, соответствующие флагам SYN, FIN, ACK заголовков пакетов. Эти позиции и образуют коммуникационную подсистему. Флаги пакетов, передаваемых правой взаимодействующей системой имеют префикс x. Отметим, что для наглядности модели флаг подтверждения ACK представлен отдельными позициями, соответствующими его получению в ответ на флаг SYN (SYNACK), либо в ответ на флаг FIN (FINACK). Кроме того, поскольку модель не содержит описаний протоколов прикладного уровня, команды OPEN, CLOSE, SEND представлены лишь в обозначениях соответствующих переходов. В качестве имён остальных переходов выбраны первые буквы ожидаемых флагов, представленные на стандартной диаграмме состояний протокола (рис. 2).

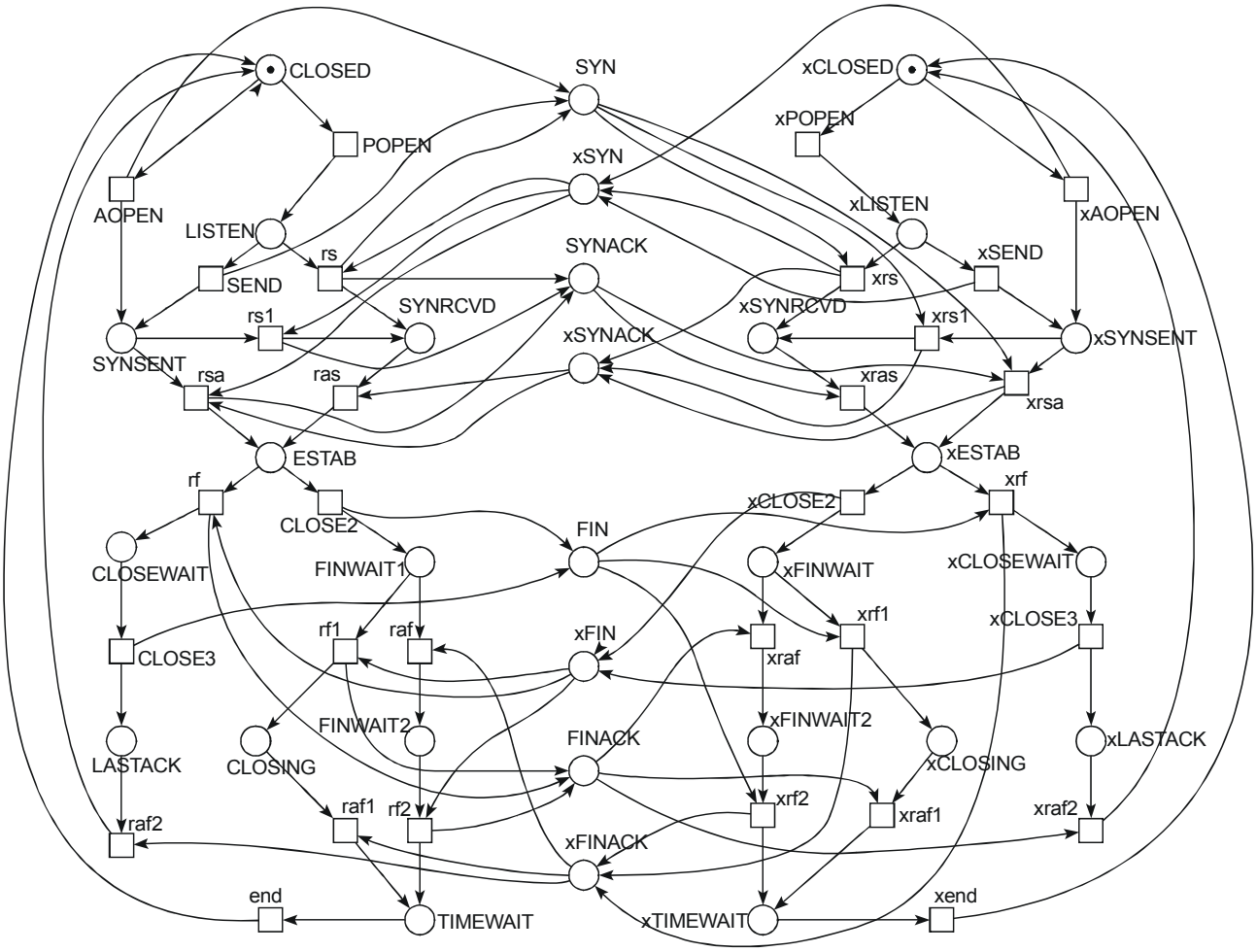


Рис. 4. Уточнённая модель Петри протокола TCP

Исследование модели, изображённой на рис. 3, показало, что сеть Петри не является ограниченной. Неограниченность сети вызвана возможностью немедленного завершения не полностью установленного соединения, например, с помощью перехода CLOSE1. Повторение последовательности AOPEN, CLOSE1 приводит к неограниченному росту маркировки позиции SYN. Такие действия, как правило, являются злонамеренными и применяются в программах, предназначенных для дестабилизации систем путём «бомбардировки» пакетами. Методы предотвращения таких ситуаций рассматривались при обсуждении средств устранения перегрузок (congestion avoidance) в FRC 896, а также при рассмотрении проблематики медленного старта в RFC 1122. В соответствии с указанными рекомендациями построена уточнённая модель протокола TCP, представленная на рис.4.

3. Декомпозиция модели

Выполним декомпозицию уточнённой модели протокола TCP, представленной на рис. 4, на минимальные функциональные подсети в соответствии с алгоритмом, описанным в [10], с помощью специально разработанной программы Deborah (www.geocities.com/zsoftua).

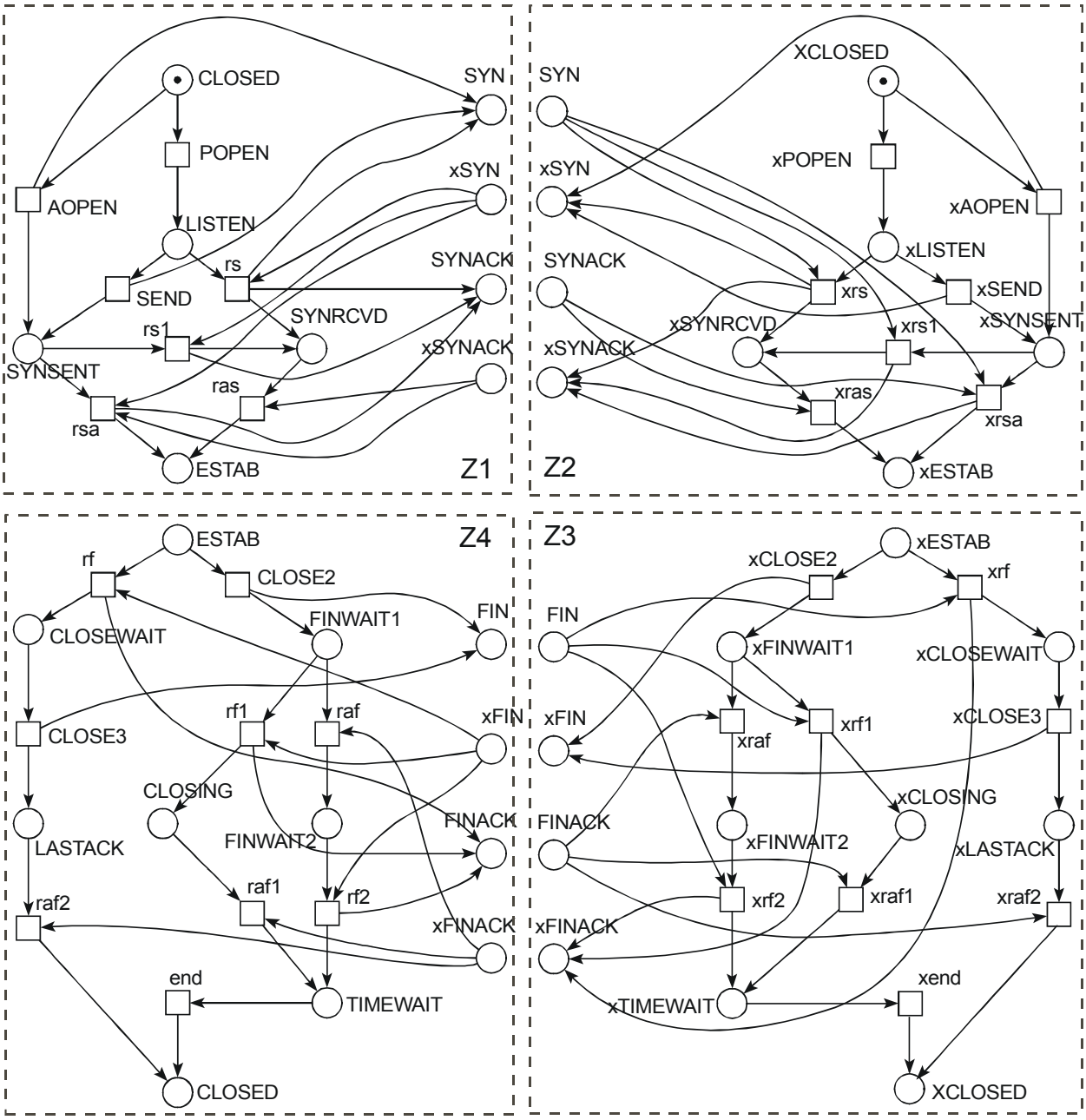


Рис. 5. Декомпозиция модели Петри протокола TCP

Применение алгоритма декомпозиции [10] к модели протокола TCP (рис. 4) приводит к получению множества $\{Z1, Z2, Z3, Z4\}$, состоящего из четырёх минимальных функциональных подсетей, представленных на рис. 5. Граф функциональных подсетей [10-15] изображён на рис. 6. Заметим, что в силу симметрии процессов взаимодействия систем, пары подсетей $Z1$ и $Z2$, а также $Z3$ и $Z4$ являются изоморфными. Поэтому в дальнейшем необходимо исследовать свойства лишь двух из перечисленных четырёх подсетей.

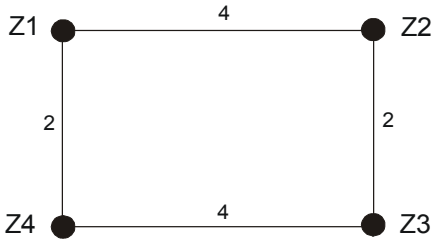


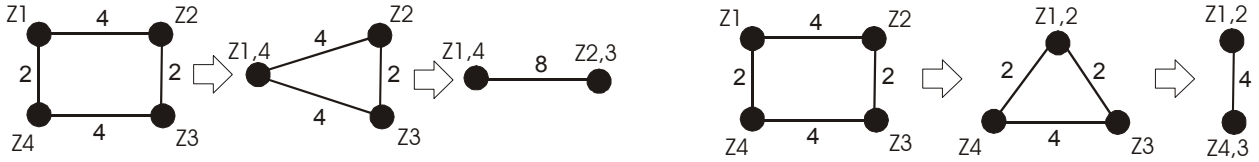
Рис. 6. Граф декомпозиции модели Петри протокола TCP

Различные способы композиция минимальных функциональных подсетей позволяет получить декомпозицию исходной модели на левую и правую взаимодействующие системы Z^L и Z^R, а также декомпозицию на сеть, устанавливающую соединение и сеть, выполняющую разъединение Z^C и Z^D, где $Z^L = Z1 + Z4$, $Z^R = Z2 + Z3$, $Z^C = Z1 + Z2$, $Z^D = Z3 + Z4$.

4. Инвариантность модели

Инварианты [3] являются мощным инструментом исследования структурных свойств сетей Петри. Они позволяют определять ограниченность, консервативность, необходимые условия живости и отсутствия тупиков. Эти свойства являются существенными для анализа поведения реальных объектов, в особенности, коммуникационных протоколов [4-7]. Модель Петри идеального телекоммуникационного протокола должна быть инвариантной [6].

Вычисление инвариантов в процессе одновременной [11] и последовательной [12,13] композиции сетей Петри из функциональных подсетей изучены на примере верификации протокола BGP [14,15]. Выполним вычисление инвариантов построенной модели Петри протокола TCP (Рис. 4) в процессе последовательной композиции функциональных подсетей. Следует отметить, что построенная модель позволяет наглядно проиллюстрировать эффективность композиционных методов. Хотя она имеет сравнительно небольшую размерность – 30 позиций и 32 переходов, вычисление инвариантов для целочисленных генераторов с помощью известной программы Tina [18] не было завершено в течение трёх суток на компьютере Pentium. Для композиционного вычисления инвариантов с помощью специально разработанной программы Adriana (www.geocities.com/zsoftua) потребовалось всего лишь десять секунд.



а) Z1+Z4; Z2+Z3; Z1,4+Z2,3 б) Z1+Z2; Z4+Z3; Z1,2+Z4,3

Рис. 7. Последовательная композиция протокола TCP

Процесс последовательной композиции применён в [17] на интуитивном уровне; при этом использована последовательность, представленная на рис. 7 а). В [12,13] задача формализована в терминах теории графов и названа оптимальным коллапсом взвешенного графа. Показано, что минимальную ширину, соответствующую размерности промежуточных систем композиции, обеспечивает коллапс, представленный на рис. 7 б). Выполним вычисление инвариантов в

процессе оптимальной последовательной композиции, изображённой на рис. 7 б) и обеспечивающей ширину коллапса, равную 4.

Занумеруем позиции сети в соответствии с табл. 1 для вычисления инвариантов. Базисные инварианты подсетей $Z1$ и $Z4$ вычислены с помощью алгоритма Тудика [8]. Инварианты изоморфных подсетей $Z2$ и $Z3$ построены из полученных инвариантов.

Таблица 1. Нумерация позиций сети

Номер	Имя	Номер	Имя	Номер	Имя
1	CLOSED	11	TIMEWAIT	21	XLISTEN
2	LISTEN	12	SYN	22	XSYNSENT
3	SYNSENT	13	XSYN	23	XSYNRCVD
4	SYNRCVD	14	SYNACK	24	XESTAB
5	ESTAB	15	xSYNACK	25	XCLOSEWAIT
6	CLOSEWAIT	16	FIN	26	xFINWAIT1
7	FINWAIT1	17	XFIN	27	XLASTACK
8	LASTACK	18	FINACK	28	XCLOSING
9	CLOSING	19	xFINACK	29	xFINWAIT2
10	FINWAIT2	20	xCLOSED	30	XTIMEWAIT

4.1. Композиция: Z1+Z2

По отношению к нумерации позиций, заданной табл. 1, инварианты подсетей $Z1$ и $Z2$ могут быть представлены как

$$(x_1, x_2, x_3, x_4, x_5, x_{12}, x_{13}, x_{14}, x_{15}) = (z_1^1, z_2^1, z_3^1, z_4^1, z_5^1, z_6^1) \cdot G^1,$$

$$(x_{20}, x_{21}, x_{22}, x_{23}, x_{24}, x_{13}, x_{12}, x_{15}, x_{14}) = (z_1^2, z_2^2, z_3^2, z_4^2, z_5^2, z_6^2) \cdot G^2,$$

где указанные матрицы имеют вид

$$G^1 = G^2 = \begin{vmatrix} 1&1&1&1&1&0&0&0&0 \\ 1&1&1&0&0&0&0&1&0 \\ 0&0&0&1&1&0&1&0&0 \\ 1&1&0&0&0&1&0&0&0 \\ 0&0&0&0&1&0&0&0&1 \\ 0&0&0&0&0&0&1&1&0 \end{vmatrix},$$

Заметим, что компоненты векторов $\bar{x}^j$, соответствующие подсетям $Z1$ и $Z2$ записаны в явной форме. Они определяют индексацию столбцов построенных матриц. Индексы строк соответствую компонентам векторов

$$\bar{z}^1 = (z_1^1, z_2^1, z_3^1, z_4^1, z_5^1, z_6^1), \bar{z}^2 = (z_1^2, z_2^2, z_3^2, z_4^2, z_5^2, z_6^2).$$

Построим систему композиции для контактных позиций:

$$\begin{cases} p_{12}: & z_4^1 - z_3^2 - z_6^2 = 0, \\ p_{13}: & z_4^2 - z_3^1 - z_6^1 = 0, \\ p_{14}: & z_2^1 + z_6^1 - z_5^2 = 0, \\ p_{15}: & z_2^2 + z_6^2 - z_5^1 = 0. \end{cases}$$

Заметим, что в композиции подсетей $Z1$ и $Z2$ использованы такие контактные позиции как p_{12}, p_{13}, p_{14}, p_{15}. Общее решение системы имеет вид

$$(z_1^1, z_2^1, z_3^1, z_4^1, z_5^1, z_6^1, z_1^2, z_2^2, z_3^2, z_4^2, z_5^2, z_6^2) = \bar{y} \cdot R^{1,2},$$

$$R^{1,2} = \begin{vmatrix} 0&0&0&0&0&0&1&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&1&0&0&0 \\ 0&0&1&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&1&0&0&0&1&1&0 \\ 0&1&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&1&1&0&0&0&0&0&0&1 \\ 0&0&0&0&1&0&0&1&0&0&0&0 \end{vmatrix}.$$

Для вычисления базисных инвариантов подсети $Z1,2$ построим объединённую матрицу $G^{1,2}$ из инвариантов подсетей G^1 и G^2:

$$G^{1,2} = \begin{vmatrix} 1&1&1&1&1&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&1&1&0&1&0&0&0&0&0&0&0&0 \\ 1&1&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&1&1&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{vmatrix}.$$

Индексация столбцов соответствует вектору

$$(x_1, x_2, x_3, x_4, x_5, x_{12}, x_{13}, x_{14}, x_{15}, x_{20}, x_{21}, x_{22}, x_{23}, x_{24}).$$

Инварианты контактных позиций вычислены в соответствии с матрицей подсети $Z1$.
Матрица базисных решений $H^{1,2} = R^{1,2} \cdot G^{1,2}$ имеет вид

$$H^{1,2} = \begin{vmatrix} 0&0&0&0&0&0&0&0&0&1&1&1&1&1 \\ 1&1&1&1&1&0&0&0&0&0&0&0&0&0 \\ 1&1&0&0&0&1&0&0&0&0&0&0&1&1 \\ 0&0&0&1&1&0&1&0&0&1&1&0&0&0 \\ 0&0&0&0&0&0&1&1&0&1&1&0&0&1 \\ 1&1&1&0&0&0&0&1&0&0&0&0&0&1 \\ 1&1&0&0&1&1&0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&1&1&1&1&0&0 \end{vmatrix}.$$

4.2. Композиция: Z4+Z3

Инварианты подсетей $Z4$ и $Z3$ могут быть представлены как

$$(x_1, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{16}, x_{17}, x_{18}, x_{19}) = (z_1^4, z_2^4, z_3^4, z_4^4, z_5^4, z_6^4) \cdot G^4,$$

$$(x_{20}, x_{24}, x_{25}, x_{26}, x_{27}, x_{28}, x_{29}, x_{30}, x_{17}, x_{16}, x_{19}, x_{18}) = (z_1^3, z_2^3, z_3^3, z_4^3, z_5^3, z_6^3) \cdot G^3,$$

где указанные матрицы имеют вид

$$G^4 = G^3 = \begin{vmatrix} 1&1&1&1&1&1&1&1&0&0&0&0 \\ 1&0&1&0&1&1&0&1&0&1&0&0 \\ 0&1&0&1&0&0&1&0&0&0&1&0 \\ 1&0&0&0&0&0&1&1&0&0&0&1 \\ 0&1&1&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&1&0 \end{vmatrix}.$$

Система композиции для контактных позиций имеет вид

$$\begin{cases} p_{16}: & z_5^4 - z_2^3 - z_6^2 = 0, \\ p_{17}: & z_5^3 - z_2^4 - z_6^2 = 0, \\ p_{18}: & z_3^4 + z_6^4 - z_4^3 = 0, \\ p_{19}: & z_3^3 + z_6^3 - z_4^4 = 0. \end{cases}$$

Общее решение может быть представлено как

$$(z_1^4, z_2^4, z_3^4, z_4^4, z_5^4, z_6^4, z_1^3, z_2^3, z_3^3, z_4^3, z_5^3, z_6^3) = \overline{y} \cdot R^{4,3},$$

$$R^{4,3} = \begin{vmatrix} 0&0&0&0&0&0&1&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&1&0&0&0&1&1&0 \\ 0&0&1&0&0&0&0&0&0&1&0&0 \\ 0&0&0&1&1&0&0&0&0&0&0&1 \\ 0&0&0&1&0&0&0&0&1&0&0&0 \end{vmatrix}.$$

Для вычисления базисных инвариантов подсети $Z4{,}3$ построим объединённую матрицу $G^{4,3}$ из инвариантов подсетей G^4 и G^3

$$G^{4,3} = \begin{vmatrix} 1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&1&0&1&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&1&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&1&1&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&1&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&1&0&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{vmatrix}.$$

Индексация столбцов соответствует вектору

$$(x_1, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{16}, x_{17}, x_{18}, x_{19}, x_{20}, x_{24}, x_{25}, x_{26}, x_{27}, x_{28}, x_{29}, x_{30}).$$

Матрица базисных решений $H^{4,3} = R^{4,3} \cdot G^{4,3}$ имеет вид

$$H^{4,3} = \begin{vmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&0&0&0&0&0&1&0&0&0&1&0&1&0&1&1&0&1 \\ 1&0&1&0&1&1&0&1&0&1&0&0&0&1&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&1&0&1&1&1&0&0&0&1&1 \\ 0&1&0&1&0&0&1&0&0&0&1&0&1&0&0&0&0&0&1&1 \\ 1&1&1&0&0&0&1&1&1&0&0&1&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&1&1&0&0&0&1&0&1&0&1&0&0&1&0 \end{vmatrix}.$$

4.3. Композиция: Z1,2 +Z4,3

Система уравнений для контактных позиций имеет вид

$$\begin{cases} p_1: & z_2^{4,3} + z_4^{4,3} + z_7^{4,3} + z_8^{4,3} - z_2^{1,2} - z_3^{1,2} - z_6^{1,2} - z_7^{1,2} = 0, \\ p_5: & z_2^{1,2} + z_4^{1,2} + z_7^{1,2} + z_8^{1,2} - z_2^{4,3} - z_3^{4,3} - z_6^{4,3} - z_7^{4,3} = 0, \\ p_{20}: & z_1^{4,3} + z_3^{4,3} + z_5^{4,3} + z_6^{4,3} - z_1^{1,2} - z_4^{1,2} - z_5^{1,2} - z_8^{1,2} = 0, \\ p_{24}: & z_1^{1,2} + z_3^{1,2} + z_5^{1,2} + z_6^{1,2} - z_1^{4,3} - z_4^{4,3} - z_5^{4,3} - z_8^{4,3} = 0. \end{cases}$$

Эта система имеет 48 базисных решений, представленных матрицей R:

$$
R = \begin{vmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0\\
0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1\\
0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0\\
1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0\\
0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1\\
0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{vmatrix}^T .
$$

Построим объединённую матрицу G из инвариантов подсетей $G^{1,2} = H^{1,2}$ и $G^{4,3} = H^{4,3}$:

$$
G = \begin{vmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&0&0&0&0&0\\
1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0\\
0&0&0&1&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&1&1&0&0&1&0&0&0&0&0&0\\
1&1&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
1&1&0&0&1&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1\\
0&0&0&0&0&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&1&1&0&1\\
0&0&0&0&0&1&0&1&1&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&0&0&0&1&1\\
0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&1\\
0&0&0&0&0&1&0&0&0&1&1&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&1&0
\end{vmatrix} .
$$

Матрица базисных решений $H^{4,3} = R^{4,3} \cdot G^{4,3}$ после удаления не минимальных решений имеет вид

$$
H = \begin{vmatrix}
1&1&1&0&0&0&0&0&0&1&1&0&0&1&0&0&0&0&1&0&0&0&0&1&0&1&0&0&1&0\\
1&1&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&1&0&0&0&1&1&0&1&0&0&1&0\\
1&1&0&0&1&1&0&0&0&1&1&1&0&0&1&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
1&1&1&1&1&1&0&0&0&1&1&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
1&1&1&0&0&1&0&1&1&0&1&0&0&1&0&0&1&0&0&0&0&0&0&1&1&0&0&0&0&0\\
1&1&0&0&0&1&0&1&1&0&1&1&0&0&0&0&1&0&0&0&0&0&1&1&1&0&0&0&0&0\\
1&1&0&0&1&1&1&1&1&1&1&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&1&0&0&1&0&0&0&0&1&0&0&1&0&1&1&1&0&0&0&0&0&0&1&1\\
0&0&0&1&1&0&1&0&0&1&0&0&1&0&0&0&0&1&0&1&1&0&0&0&0&0&0&0&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&1&0&1&1&0&0&1&1&0&0&0&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&1&1&1&1&1&1&0&0&0&1&1\\
0&0&0&0&1&1&0&0&0&0&0&0&0&0&1&1&0&0&0&1&1&1&0&0&1&0&1&1&0&1\\
0&0&0&1&1&1&0&0&0&0&0&0&1&0&0&1&0&0&0&1&1&0&0&0&1&0&1&1&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&1&1&0&0&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1\\
1&1&0&0&1&0&1&0&0&2&1&1&1&1&1&0&0&1&1&1&1&0&0&1&0&1&0&0&2&1\\
1&1&1&1&1&0&1&0&0&2&1&0&0&0&0&0&0&1&1&1&1&1&1&1&0&1&0&0&2&1\\
1&1&0&0&1&2&0&1&1&0&1&1&1&1&1&1&1&0&0&1&1&0&0&1&2&0&1&1&0&1\\
1&1&1&1&1&2&0&1&1&0&1&0&0&0&0&1&1&0&0&1&1&1&1&1&2&0&1&1&0&1\\
1&1&1&0&1&1&0&0&0&1&1&0&0&1&1&1&1&1&1&1&1&1&0&1&1&0&0&0&1&1\\
1&1&0&1&1&1&0&0&0&1&1&1&1&0&0&1&1&1&1&1&1&1&0&1&1&1&0&0&1&1\\
1&1&1&0&1&1&1&1&1&1&1&0&0&1&1&0&0&0&0&1&1&1&0&1&1&1&1&1&1&1\\
1&1&0&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&1&1&0&1&1&1&1&1&1&1&1
\end{vmatrix} .
$$

Так как, например, сумма всех строк содержит все натуральные компоненты, модель протокола является p-инвариантной и, следовательно, сеть является ограниченной и консервативной. Аналогичным образом [13] с использованием двойственной сети можно показать, что она является также и t-инвариантной; это означает, что сеть Петри стационарно повторяемая и представляет необходимые условия её живости.

5. Анализ пространства состояний

Для исследования свойств модели протокола TCP применена также стандартная методика, использующая граф достижимых маркировок сети Петри [3]. Граф достижимых маркировок, представленный на Рис. 8 содержит 58 вершин и позволяет определить ограниченность и

живость модели протокола с помощью метода полного перебора. Граф достижимых маркировок (Рис. 8) подтверждает достоверность инвариантов, полученных с помощью декомпозиции модели. Описание маркировок графа приведено в Таблице 2.

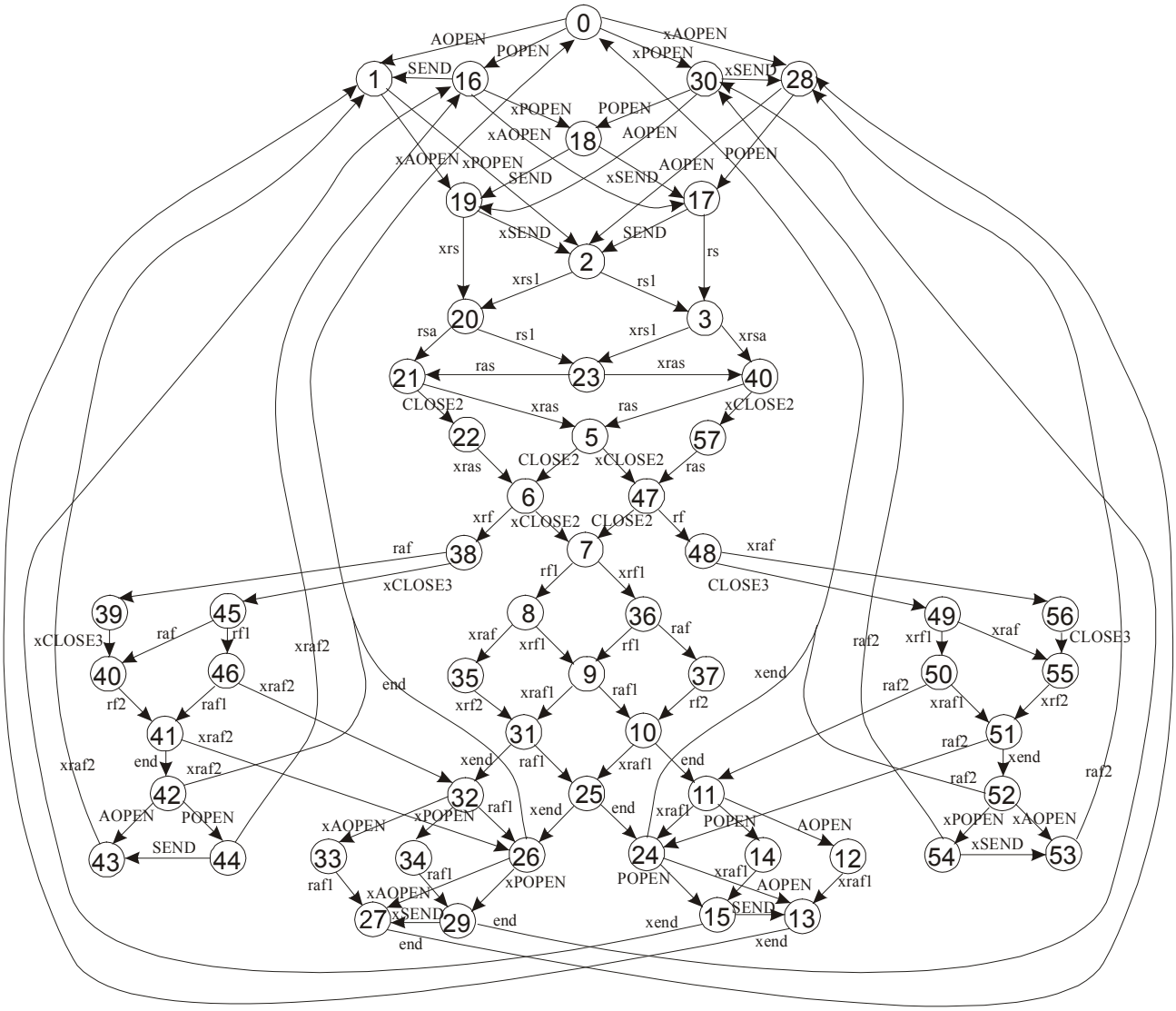


Рис. 7. Граф достижимых маркировок модели Петри протокола TCP

Используем граф для проверки свойств полученных инвариантов. Известно [3], что инвариант позиций представляет собой вектор весов, такой, что взвешенная сумма фишек остаётся постоянной в любой достижимой маркировке сети. Например, для инварианта

```
(2 2 1 1 2 2 1 1 1 2 2 1 1 1 1 1 1 1 2 2 1 1 2 2 1 1 1 2 2)
```

взвешенная сумма фишек в начальной маркировке равняется 4 и остаётся таковой в любой из достижимых маркировок, приведенных в Таблице 2.

Таблица 2. Описание маркировок сети Петри

№	Маркировка	№	Маркировка
0	CLOSED xCLOSED	29	TIMEWAIT xLISTEN
1	SYN SYNSENT xCLOSED	30	CLOSED xLISTEN
2	SYN SYNSENT xSYN xSYNSENT	31	CLOSING xFINACK xTIMEWAIT
3	SYN SYNACK SYNRCVD xSYNSENT	32	CLOSING xCLOSED xFINACK
4	SYNRCVD xESTAB xSYNACK	33	CLOSING xFINACK xSYN xSYNSENT
5	ESTAB xESTAB	34	CLOSING xFINACK xLISTEN
6	FIN FINWAIT1 xESTAB	35	CLOSING FIN xFINWAIT2
7	FIN FINWAIT1 xFIN xFINWAIT1	36	FINWAIT1 xCLOSING xFIN xFINACK
8	CLOSING FIN FINACK xFINWAIT1	37	FINWAIT2 xCLOSING xFIN
9	CLOSING FINACK xCLOSING xFINACK	38	FINWAIT1 xCLOSEWAIT xFINACK
10	FINACK TIMEWAIT xCLOSING	39	FINWAIT2 xCLOSEWAIT
11	CLOSED FINACK xCLOSING	40	FINWAIT2 xFIN xLASTACK
12	FINACK SYN SYNSENT xCLOSING	41	FINACK TIMEWAIT xLASTACK
13	SYN SYNSENT xTIMEWAIT	42	CLOSED FINACK xLASTACK
14	FINACK LISTEN xCLOSING	43	FINACK SYN SYNSENT xLASTACK
15	LISTEN xTIMEWAIT	44	FINACK LISTEN xLASTACK
16	LISTEN xCLOSED	45	FINWAIT1 xFIN xFINACK xLASTACK
17	LISTEN xSYN xSYNSENT	46	CLOSING FINACK xFINACK xLASTACK
18	LISTEN xLISTEN	47	ESTAB xFIN xFINWAIT1
19	SYN SYNSENT xLISTEN	48	CLOSEWAIT FINACK xFINWAIT1
20	SYNSENT xSYN xSYNACK xSYNRCVD	49	FIN FINACK LASTACK xFINWAIT1
21	ESTAB SYNACK xSYNRCVD	50	FINACK LASTACK xCLOSING xFINACK
22	FIN FINWAIT1 SYNACK xSYNRCVD	51	LASTACK xFINACK xTIMEWAIT
23	SYNACK SYNRCVD xSYNACK xSYNRCVD	52	LASTACK xCLOSED xFINACK
24	CLOSED xTIMEWAIT	53	LASTACK xFINACK xSYN xSYNSENT
25	TIMEWAIT xTIMEWAIT	54	LASTACK xFINACK xLISTEN
26	TIMEWAIT xCLOSED	55	FIN LASTACK xFINWAIT2
27	TIMEWAIT xSYN xSYNSENT	56	CLOSEWAIT xFINWAIT2
28	CLOSED xSYN xSYNSENT	57	SYNRCVD xFIN xFINWAIT1 xSYNACK

Инвариант переходов [3] представляет собой вектор счёта стационарно повторяющихся последовательностей срабатывания переходов. Наличие таких последовательностей является необходимым условием живости сети. С помощью графа достижимых маркировок (Рис. 8) можно установить, что, например, последовательность

```
AOPEN xPOPEN xrs rsa xras CLOSE2 xrf raf xCLOSE3 rf2 xraf2 end POPEN xAOPEN rs
xrsa res xCLOSE2 rf xraf CLOSE3 xrf2 raf2 xend POPEN SEND xPOPEN SEND rs1 xrs1
ras xras CLOSE2 xCLOSE2 rf1 xrf1 raf1 xraf1 end xend
```

является стационарно повторяющейся и содержит все переходы сети Петри. Она соответствует t-инварианту

```
AOPEN POPEN*2 SEND rs rs1 rsa ras*2 xAOPEN xPOPEN*2 xSEND xrs xrs1 xrsa xras rf
CLOSE2*2 rf1 raf CLOSE3 raf1 rf2 raf2 end*2 xrf xCLOSE2*2 xrf1 xraf xCLOSE3
xraf1 xrf2 xraf2 xend*2
```

Таким образом, выполнение свойства инвариантов, полученных в процессе последовательной композиции модели протокола TCP из функциональных подсетей, подтверждено с помощью графа достижимых маркировок.

6. Выводы

Основным результатом настоящей работы является формальное доказательства корректности процедур установления соединения и разъединения протокола TCP. Для достижения этой цели построена модель протокола в форме сети Петри с уровнем детализации, представляющим флаги стандартного заголовка сообщения. Инвариантность модели определена в процессе её последовательной композиции из функциональных подсетей, обеспечивающем существенное ускорение вычислений. Полученные результаты подтверждены также с помощью графа достижимых маркировок сети Петри.

Литература

0. *Postel, J., Editor, "Transmission Control Protocol," STD 7, RFC 793, September 1981.*
0. *Russell T. Telecommunications Protocols, 2nd Edition, McGraw-Hill, 2004.*
0. *Murata T. Petri Nets: Properties, Analysis and Applications // Proceedings of the IEEE, April 1989.- Vol. 77.- p. 541-580.*
0. *Diaz M. Modelling and Analysis of Communication and Cooperation Protocols Using Petri Net Based Model // Computer Networks.- no 6.- 1982.- p. 419-441.*
0. *Berthelot G., Terrat R. Petri Nets Theory for the Correctness of Protocols // IEEE Trans. on Communications, no. 12, 1982. Vol. 30, p. 2497-2505.*
0. *Girault C., Volk R. Petri nets for systems engineering – A guide to modelling, verification and applications. Springer-Verlag, 2003.*
0. *Cortadella J., Kishinevsky M., Kondratyev A., Lavagno L., Yakovlev A. Logic synthesis of asynchronous controllers and interfaces. Springer-Verlag, 2002.*
0. *Toudic J.M. Linear Algebra Algorithms for the Structural Analysis of Petri Nets // Rev. Tech. Thomson CSF, 1982.- No. 1.- Vol. 14.- p. 136-156.*
0. *Крывый С.Л. О некоторых методах решения и критериях совместимости систем линейных диофантовых уравнений в области натуральных чисел // Кибернетика и системный анализ, 1999, № 4, с. 12-36.*
0. *Зайцев Д.А. Декомпозиция сетей Петри // Кибернетика и системный анализ, №5, 2004, с. 131-140.*
0. *Зайцев Д.А. Инварианты функциональных подсетей // Научные труды Одесской национальной академии связи им. А.С.Попова, №4, 2003, с. 3-11.*
0. *Зайцев Д.А. Последовательная композиция функциональных подсетей // Научные труды Одесской национальной академии связи им. А.С.Попова, № 3, 2004, с. 33-40.*
0. *Zaitsev D.A. Functional Petri Nets, Universite Paris-Dauphine, Cahier du Lamsade 224, Avril 2005, 62p (www.lamsade.dauphine.fr/cahiers.html).*
0. *Зайцев Д.А. Верификация телекоммуникационных протоколов с помощью декомпозиции сетей Петри // Зв'язок №1(53), 2005, с. 41-47.*
0. *Зайцев Д.А. Последовательная композиция моделей Петри телекоммуникационных протоколов // Зв'язок, в печати.*
0. *Слепцов А.И., Юрасов А.А. Автоматизация проектирования управляющих систем гибких автоматизированных производств / Под ред. Б.Н.Малиновского. К.: Техніка, 1986, 160 с.*
0. *Зайцев Д.А. Инвариантность модели Петри протокола TCP // Научные труды Одесской национальной академии связи им. А.С.Попова, № 2, 2004, с. 19-27.*
1. *Berthomieu B., Ribet O.-P., Vernadat F. The tool TINA - construction of abstract state space for Petri nets and Time Petri nets // International Journal of Production Research, Vol. 42, no. 4, 2004 (http://www.laas.fr/tina).*