

Зимин Андрей Владимирович
старший преподаватель кафедры ИКС
СПбГУТ им. проф. М.А. Бонч-Бруевича”

МОДЕЛИ И МЕТОДЫ ПРОВЕДЕНИЯ ТЕСТОВ ТЕЛЕКОММУНИКАЦИЙ 5G/6G С СЕТЯМИ ПЕТРИ

Аннотация. В мультисервисных сетях Beyond-5G (B5G) предполагаются множество новых услуг в реальном времени и беспрецедентный рост числа подписок на эти услуги. Все это требует дополнительных значительных усилий по тестированию новых телекоммуникационных протоколов и сценариев взаимодействия сетевых элементов сетей B5G. Новые научные методы тестирования с привлечением аппарата сетей Петри рассматриваются в этой статье на примере протокола SIP

Ключевые слова: Протоколы сигнализации, тестирование сетевой инфраструктуры, SIP, сети Петри, поколения Beyond-5G

Постановка задачи.

В общем случае тестирование телекоммуникационных протоколов – это разновидность тестирования черного ящика, при котором проверяется соответствие конкретной реализации протокола его текстовому описанию, приведенному в стандартах и сценариях взаимодействия этих протоколов. Такие испытания являются основой для многих других тестов. При этом сети Петри позволяют использовать методы графического представления и строгого математического определения, что делает работу по проверке соответствия протокола более точной и предоставляет теоретическую и математическую основу для испытаний.

В общем виде можно выделить следующие этапы тестирования на соответствие, которые будут верны для любого протокола с установлением сеанса: подготовка к тестированию; проведение теста; выполнение теста; выводы по итогам тестирования.

Рассмотрим эти этапы на примере протокола SIP, причем каждый этап и каждую часть теста интегрируем в формализованную среду сети Петри. Рабочий процесс такого тестирования представлен на рис.1.

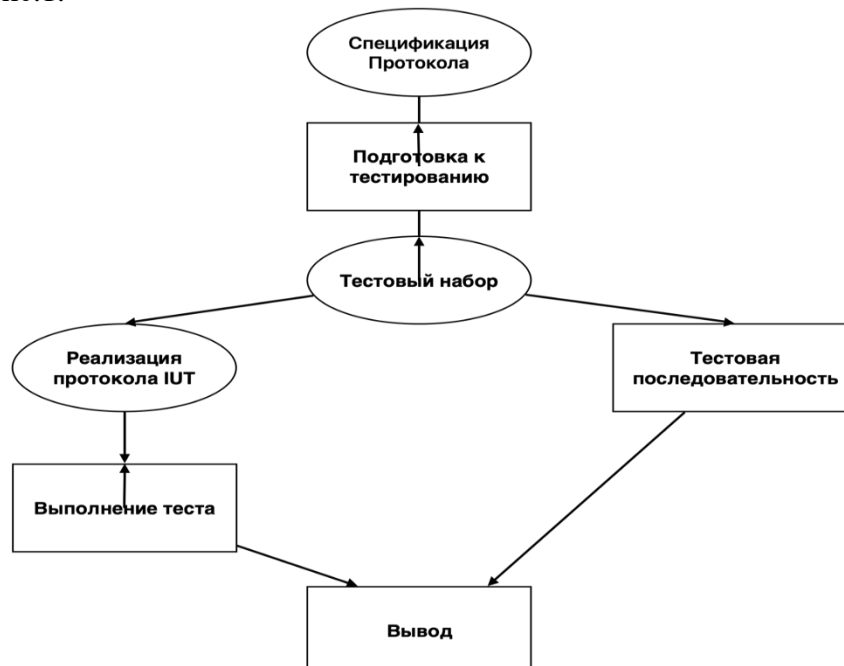


Рис. 1. Общая схема тестирования на соответствие

Подготовка к тестированию включает исследование тестирования протокола, сфокусированное на том, как получить из описания протокола необходимый тестовый набор (тест-кейс). Тестовый набор (тест-кейс) представляет собой набор, который обеспечивает для каждого тестового случая наименьшее возможное количество тестовых сценариев, охватывающее при этом как можно большее количество ситуаций, возникающих в процессе выполнения протокола. Тестовая последовательность предназначена для получения полной ожидаемой последовательности ответов тестовых сценариев. Выполнение теста заключается в получении последовательности ответов от тестовых сценариев в результате их реализации посредством IUT (протокола, предназначенного для тестирования), генерации файлов и создания журнала тестирования. Выводы по итогам тестирования делаются после того, как записи тестов проанализированы и сравнены с ожидаемой последовательностью ответов на тестовые примеры.

Рассмотрим несколько подробнее архитектура тестирования на соответствие протоколу, представленную на рис.2.

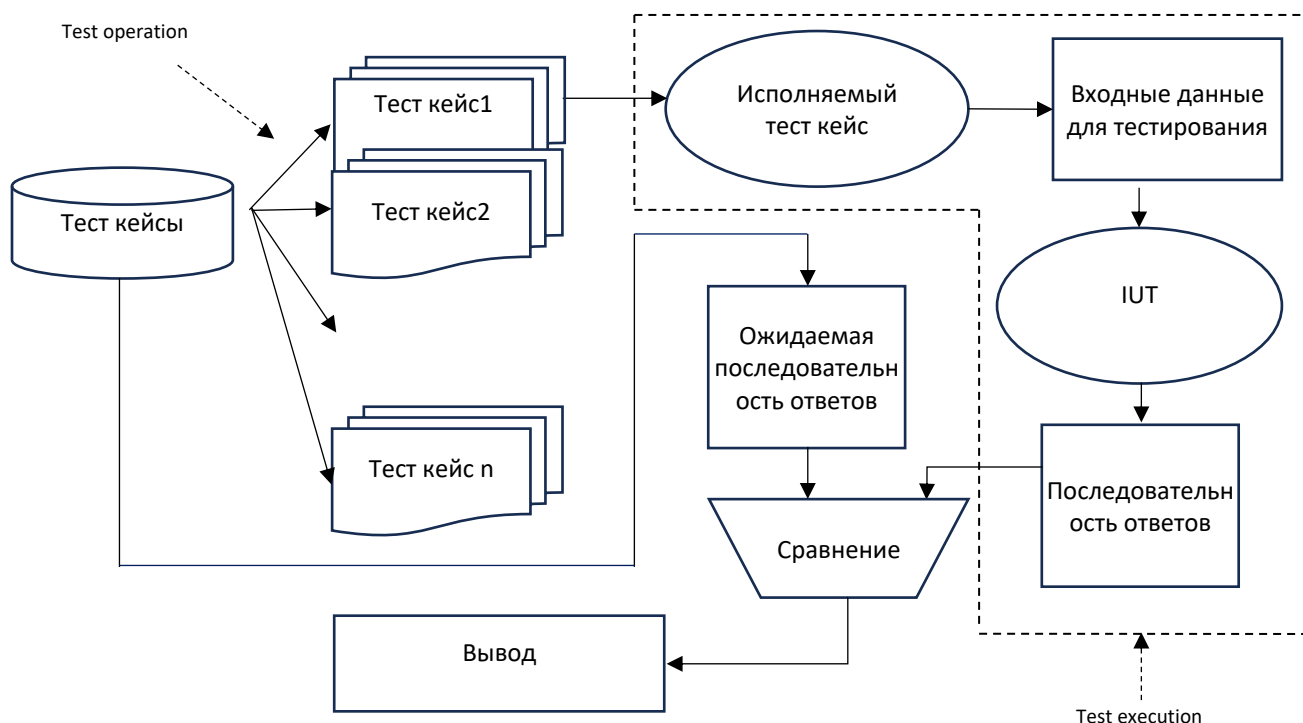


Рис.2. Метод проведения тестов с сетями Петри, исполнения и оценки

На первом этапе выполняется подготовки к тестированию, текстовое описание рассматривается при подготовке к тестированию с учетом назначения протокола, который должен быть протестирован, и приводится соответствующий набор тестов. Как правило, для проведения функционального тестирования или части теста протокола требуется более одного тестового сценария, для того чтобы охватить все возможные условия работы протокола. В дополнение к КАП и MSC-сценариями тестовый набор выражается моделью сети Петри для этого же протоколу, сетевая система тестовых сценариев также выражается процессом с применением сети Петри.

После этапа подготовки теста, получения набора тестов и тестового сценария, тестовая работа делится на два связанных этапа. Одним из шагов является генерация исполняемых тестовых сценариев. Для каждого исполняемого тестового сценария задаются начальные условия (например, значения параметров, начальное значение счетчика таймера и т.д.).

Каждый исполняемый тестовый сценарий передается в IUT (тестируемую реализацию) для выполнения. Результатом ответа IUT является единица данных протокола PDU (Protokol Data Unit), которая записывает поведение IUT при выполнении исполняемого тестового сценария. PDU также может формировать файлы журнала тестирования. Вторым шагом является получение ожидаемой последовательности ответов. Этот процесс заключается в анализе работы сети Петри. Эта работа имеет реальный физический смысл, т.е. мы можем установить соответствующую взаимосвязь между процессом работы сети Петри и реальным поведением IUT. Наконец, фактическая последовательность ответов протокола сравнивается с ожидаемой последовательностью ответов, чтобы определить, соответствует ли поведение объекта протокола текстовому описанию протокола.

Моделирование протоколов средствами сетей Петри

Теперь подробнее рассмотрим моделирование протокола с применением сети Петри. В диссертационной работе мы будем использовать метод автоматического моделирования сети Петри, основанный на последовательности поведения системы. Как именно использовать этот метод для создания модели сети Петри протокола, будет описано ниже.

Как показано на рис.3 в методе используются следующая идея – функциональное описание модели сети Петри можно рассматривать в виде абстрактного тестового набора для тестирования того или иного сетевого элемента, который моделирует все возможные режимы работы. Если задана начальная маркировка, то задается язык сети Петри для модели, что эквивалентно знанию ожидаемой последовательности и рабочего состояния всех протоколов. Любая строка, принятая сетью Петри, может отражать поведение протокола. Строка может быть использована в качестве процедуры ожидаемой последовательности ответов в тестовом сценарии. В соответствии с первоначальной маркировкой сети Петри и меткой *положения*, мы можем генерировать исполняемые тестовые сценарии. Исполняемые тестовые сценарии будут выполнены IUT. Результатом ответа IUT является PDU (блок протокольных данных), который записывает результаты своей работы.

Таким образом, любое принятое сетью Петри значение σ может быть выражено в виде строки, которая отражает определенное поведение протокола. Первоначальная идентификация на основе сети Петри и аннотация библиотек могут генерировать исполняемые тестовые сценарии. Запускаем исполняемые тестовые сценарии в IUT (тестируемую реализацию), документируя результаты ее работы. Затем мы преобразуем строку текущих результатов σ в последовательности *переходов* сети Петри. Если σ может быть принято сетью Петри протокола и после $M_0[\sigma > M'$, M' для сопоставления с конечным желаемым состоянием протокола, то испытание прошло успешно. В противном случае, если M' не соответствует конечному желаемому состоянию протокола, пусть M_0 будет начальной разметкой модели сети Петри, запускаем последовательность *переходов* σ , $M_0[\sigma > M''$. Очевидно, M'' – это ошибка в состоянии выполнения протокола. Наконец, по результатам тестирования формируется протокол тестирования.

Сеть Петри для INVITE

Для того чтобы проиллюстрировать предложенный метод моделирования, рассмотрим его для протокола прикладного уровня SIP (Session Initiation Protocol) [3]. Используя метод



моделирования, приведем модель сети Петри для протокола SIP, рис.3. Ниже приведены пояснения к местам и переходам в модели SIP. Используя модель сети Петри, соответствие протокола SIP может быть проверено в различных сценариях SIP. Этот пример также является дальнейшим развитием спецификации тестирования протокола SIP.

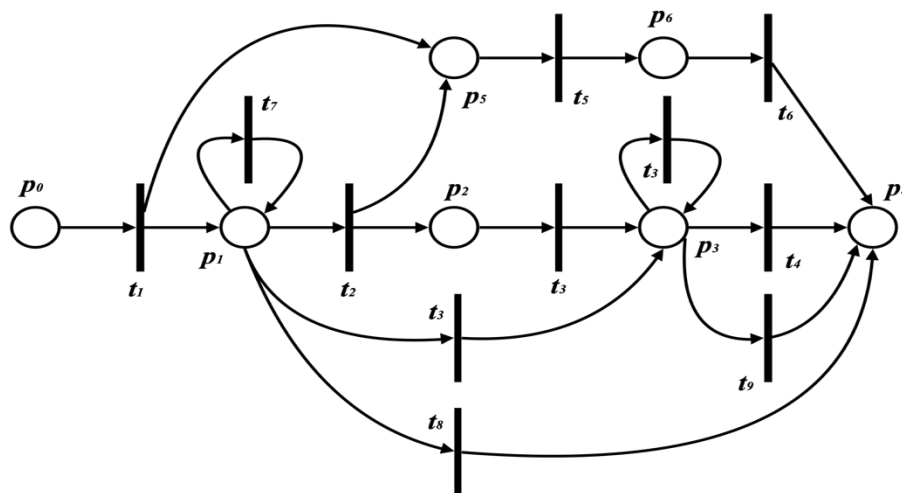


Рис.3. Сеть Петри для SIP INVITE со стороны клиента

Данная сеть состоит из следующих элементов:

Позиции:

P0 – Pre-Calling State

P1 – Отправлен INVITE, Calling State

P2 – Получен промежуточный ответ 1, терминал перешел в Proceeding State

P3 – Отправить ACK, перейти к complete State

P4 – Завершение транзакции

P5 – Состояние ANY Ожидание ответа

P6 – Получил промежуточный ответ 2, готовность к отправке ACK

Переходы:

t1 – отправка INVITE

t2 – Получен промежуточный ответ 1

t3 – Получен финальный ответ

t4 – timer D, overtime

t5 – Получен промежуточный ответ

t6 – отправка ACK

t7 – сработал таймер ожидания ответа, перепосылка INVITE

t8 – сработал таймер В по неответу, ошибка передачи

t9 – ошибка передачи

В общем случае терминал клиента может находиться в 4 состояниях:

- Pre-Calling,
- Calling,
- Proceeding,
- Completed.

Изначально устройство находится в состоянии Pre-calling. Отправив запрос на установление соединения (отправка INVITE, начало транзакции), устройство переходит в состояние Calling. В то же время клиентская транзакция передается на транспортный уровень для передачи по сети. Если транзакция находится в состоянии Calling state, а таймер В истек,

участники транзакции должны быть об этом уведомлены. Уровень транзакций не может генерировать АСК. Если участник транзакции получает ответ в состоянии Calling, статус изменяется и переходит в состояние Proceeding (обработки). Независимо от того, находится ли транзакция в состоянии Calling (вызова) или в состоянии Proceeding (обработки), при получении ответа состояние транзакции переходит в состояние Completed (Завершено). При этом должен быть сгенерирован соответствующий АСК и отправлен на транспортный уровень. Добавим к этому, что в начале запуска протокола SIP, в позиции p_0 есть маркер, а остальные позиции не содержат маркеров. Таким образом, первоначальная маркировка модели сети Петри такова $M_0(P_0) = 1$ и также $M_0(P_1) = M_0(P_2) = M_0(P_3) = M_0(P_4) = M_0(P_5) = M_0(P_6) = 1$. Значение таймера В устанавливаем равным 200 мс.

Тестовый набор

С учетом этого рассмотрим генерацию тестового набора и тестовых сценариев протокола SIP. В соответствии с моделью сети Петри протокола SIP на рис.3.9 мы можем получить тестовый набор и тест-кейсы. Собственно говоря, модель сети Петри на рис.3.9 уже можно рассматривать как тестовый набор. Протокол SIP имеет 3 режима работы, соответствующих трем тестовым сценариям. Тестовые сценарии в сетевой системе приведены на рис.3.10. *Case1*, *Case2*, *Case3* – это тестовые сценарии, которые выражают все возможные варианты поведения ветви протокола SIP. Смысл каждого тестового сценария заключается в следующем:

Case1 представляет собой следующую процедуру тестирования. Протокол SIP переходит в состояние trying (попытки), когда клиентский терминал генерирует новый экземпляр транзакции. В то же время, если таймер истек, запрашивается повторная передача. Если клиентский терминал получает подтверждение, SIP переходит в состояние proceeding (обработка) и уведомляет транспортный уровень. Если клиентский терминал получает окончательный ответ, статус SIP переводится в состояние completed "завершено".

Case 2 – клиентский терминал может отправлять более одного ответа на транзакции на стороне сервера. В состоянии proceeding транзакция отправляет эти ответы на транспортный уровень для передачи по сети, но не влияет на состояние. Если запрос на повторную передачу получен в этом состоянии, транзакция повторно передаст самый последний ответный запрос Ask. Если ответ на кадр данных 2 получен в этом состоянии (состояние proceeding – обработка), экземпляр транзакции отправляет кадр данных 2 на транспортный уровень для передачи, но не несет ответственности за повторную передачу.

Case3 – для запроса без приглашения исходным состоянием протокола SIP является состояние ожидания. После перехода в состояние ожидания любой полученный запрос на повторную передачу будет отклонен. В этом состоянии, если ответ получен с верхнего уровня, сервер перейдет в состояние proceeding (обработки). Если ответ клиентского терминала получен серверным терминалом в этом состоянии, протокол SIP переводится в состояние completed (завершено). Одновременно отправляется ответ на транспортный уровень для его дальнейшей передачи.

Заключение

В статье предложена схема метода тестирования соответствия сетевого протокола, набор тестов протокола и метод генерации тестовых примеров на базе сети Петри. Создана модель тестирования на соответствие протокола SIP. Показано, что предложенный метод делает сеть Петри настоящей моделью формализации на протяжении всего процесса тестирования соответствия протоколу, а абстрактный набор тестов, тестовый пример, последовательность



тестов – все это выражается сетью Петри. Разработан метод тестирования мультисервисных сетей поколений B5G на базе цифровых близнецов, позволяющий операторам справляться с новыми вызовами, проводить тщательный анализ текущих тенденций и быть готовыми принимать проактивные решения в режиме ЧЕС-тестирования (по «что-если-сценариям») без реального вмешательства в работу сети

Список литературы:

1. Б. С. Гольдштейн, А. В. Зимин. Сдвиг парадигмы тестирования сетевой инфраструктуры 5G/6G. «Вестник связи», № 2, 2025
2. H. Wang, Y. Wu, G. Min, W. Miao. A graph neural network-based digital twin for network slicing management. IEEE Trans. Ind. Informat., vol. 18, no. 2, pp. 1367_1376, Feb. 2022.
3. Гольдштейн А.Б., Зимин А.В. Специфика сертификационных испытаний OSS/BSS. «Вестник связи» №12/2020.
4. R. Kerris. (2023). Ericsson Builds Digital Twins for 5G Networks in Nvidia Omniverse [Online]: <https://blogs.nvidia.com/blog/2021/11/09/ericsson-digital-twins-omniverse/>

