

Методология воспроизводимых исследований имитационных моделей и реализаций коммуникационных протоколов

Д.Ю. Чалый

Ярославский государственный университет

им. П.Г. Демидова

Ярославль

Email: chaly@uniyar.ac.ru

Abstract—Транспортные протоколы, в особенности протокол TCP (Transmission Control Protocol), являются ключевым компонентом архитектуры TCP/IP, используемой в современных коммуникационных сетях. Разработка и исследование свойств новых транспортных протоколов требует умения решать широкий спектр задач в области их моделирования, верификации и проведении анализа производительности. Однако зачастую в рамках этих исследований еще одна задача остается нерассмотренной — организация процесса исследований таким образом, чтобы они могли быть автоматизировано, а в идеале автоматически, воспроизведены другими учеными. В этой статье рассматриваются подходы и методики, которые позволяют обеспечить воспроизводимость результатов исследований (reproducible research) в области коммуникационных протоколов.

I. Введение

Воспроизводимость результатов экспериментальных научных исследований независимыми научными группами было и будет важным критерием истинности сделанных выводов. Такой критерий истинности применяется в ряде научных дисциплин, например, в физике и медицине. Однако в информатике в целом, и в частности в области коммуникационных протоколов этому не уделяется должного внимания. В частности, такое мнение отражено в [1], это же подтверждает личный опыт автора этой статьи. Хотя воспроизводимость является важной характеристикой исследования и ее важность была подчеркнута Кнутом [7], и другими учеными [8], [9], [10].

Идея воспроизводимых исследований (reproducible research) в области коммуникационных протоколов давно интересует научное сообщество. Действительно, коммуникационные протоколы являются ключевым компонентом архитектуры сети Интернет, научный интерес к этой тематике не угасает, однако до сих пор трудной задачей является разработка нового протокола и проведение сравнительного анализа производительности нескольких протоколов и без того, чтобы исследование было воспроизводимым, даже несмотря на то, что за последние 20 лет были вложены колоссальные усилия в разработку средств моделирования компьютерных сетей.

II. Представление результатов исследований

В идеале научная статья должна компилироваться из набора исходных текстов, программного кода и исходных данных, однако современные технические средства и используемые методологии проведения исследований не позволяют это делать в полной мере. Решающим схожие цели является подход, который называется грамотное программирование (literate programming), предложенный Кнутом в [7]. В рамках этого подхода программирование ведется наряду с документированием кода, и из одного и того же файла может получаться и документация, и исполняемый файл. Сейчас такой подход является общепринятым в индустрии производства программного обеспечения, существуют системы документирования исходных кодов, например Doxygen [15]. Однако при проведении научных исследований такого подхода не применяется. Действительно, исследование не является программой, и если даже согласиться с точкой зрения, что все его шаги должны быть задокументированы, то совершенно непонятно как это сделать. Для лучшего понимания как это сделать необходимо формализовать шаги процесса исследования чтобы предложить варианты решений этой проблемы. Эта статья ограничивается рассмотрением лишь узкого класса исследований, а точнее, исследования в области коммуникационных протоколов, и не претендует на всеобщность, хотя аналогичные усилия могут быть применены для других научных областей.

Действительно, к анализу научных результатов автор и его коллега-читатель подходят с разных сторон. Схематично научное исследование можно представить так, как показано на рис. 1, что было предложено в [14] в контексте исследований в области наук о данных (data science). Так, автор движется в этой схеме слева направо, а читатель в обратном порядке — справа налево. При этом очевидно, что автор должен проделать весь путь, чтобы опубликовать свою работу, в то время как читатель может остановиться на любом шаге, и это зависит не только от его интереса, но и от возможности доступа к отдельным элементам исследования.

Рассмотрим эти элементы подробнее. Исходные

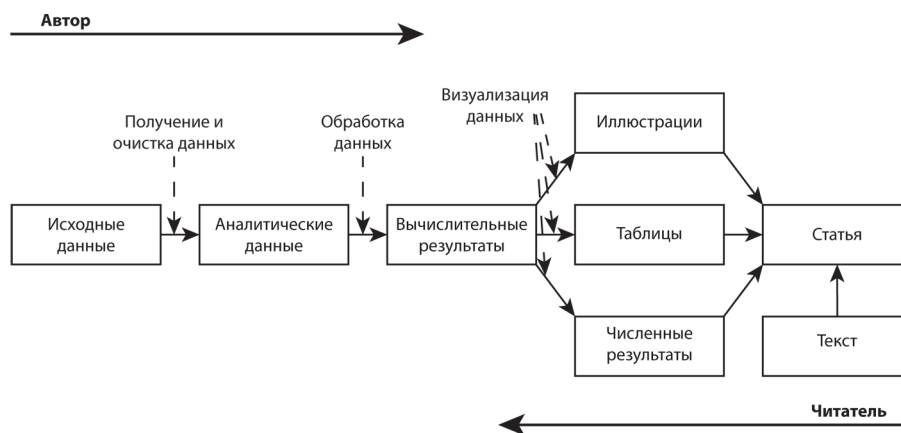


Figure 1. Процесс исследований с точки зрения автора и читателя

данные представляют собой такие данные, которые непосредственно регистрируются на выходе от источника данных. Таким источником может быть прибор, модель, или какой-то другой генератор данных. Исходные данные могут содержать ошибки, неточности, быть записанными в произвольном, или даже нескольких разных форматах. Такие данные скорее всего не подходят для дальнейшего анализа и их необходимо привести к однородному представлению, которое называется аналитическими данными и допускает их дальнейший анализ.

Аналитические данные представляются в виде таблиц, где столбцы представляют различные переменные, а каждое измерение находится в отдельной строке. Аналитические данные получают из исходных данных путем программной обработки, которая может включать трансформацию данных, выделение необходимых подмножеств данных, слияние данных, вставку данных согласно некоторой модели и т.д. Собственно наиболее подходящий способ получить аналитические данные — написать программу, которая преобразует исходные данные в аналитические.

Вычислительные результаты получают на основе аналитических данных, например, при помощи соответствующих программных средств [13] и далее иллюстрируются, резюмируются и представляются в виде таблиц, которые в итоге ложатся в основу научной статьи, которая представляет полученный научный результат.

Воспроизводимость исследований не всегда является достижимой. Препятствиями могут быть недостаток средств и вычислительных ресурсов для повторного проведения исследований, а также уникальный характер самого исследования. Например, в области транспортных протоколов мы проводили исследование [24], в рамках которого обрабатывалось порядка 0,5 Тб исходных данных, генерированных в общей сложности несколькими сотнями экспериментов на одной топологии коммуникационной сети. Примером уникального по своему характеру исследования может служить исследование [25], где анализировались характеристики сети Интернет, которые были присущи

ей в конце 1990-х годов. Очевидно, что сейчас сеть Интернет может иметь совсем другие характеристики, отличные от тех, что были тогда, более того, ее не вернуть в то состояние чтобы проверить результаты заново.

Проведение воспроизводимых исследований упрощает передачу знаний между научными коллективами и увеличивает уровень достоверности проведенных исследований. За последние годы было проведено значительное количество экспериментальных исследований, однако зачастую тот, кто знакомится с ними, не обладает средствами проверки результатов этих исследований, либо средствами опробования своих идей на основе этих публикаций. Воспроизводимость является важной характеристикой, которая должна стать стандартом де-факто любых экспериментальных исследований в области коммуникационных протоколов. Исследование, проведенное в рамках такой методологии, должно предоставлять всем желающим исходный код моделей протоколов, полученные экспериментальные данные, код, который обрабатывает эти данные и генерирует результаты.

III. Методологии исследований в области коммуникационных протоколов

В предыдущем разделе мы попытались представить, как должен быть организован процесс воспроизводимых исследований. Попробуем разобраться с методологиями, которые используются в настоящее время в области коммуникационных протоколов. Наш опыт показывает, что их можно разделить на несколько классов.

Во-первых, это исследования, в которых предлагается математическая модель телекоммуникационной системы и которая исследуется при помощи математического аппарата. Этот вид исследований не всегда имеет экспериментальный характер. Примерами таких статей являются исследования в области самоподобия телекоммуникационного трафика [16], уравнение для моделирования поведения протокола TCP Reno [17].

В этом случае представляется целостный результат, корректность которого может быть проанализирована при помощи математического аппарата вручную при наличии достаточного детальных доказательств, что и является воспроизведением таких исследований. Недостатком такого подхода является то, что воспроизвести исследование можно только вручную, а применимость средств для автоматизированного построения доказательств не является распространенной практикой в настоящее время, хотя есть работы, которые утверждают, что такой подход может стать стандартом в перспективе ближайших десятилетий [18]. Далее мы не будем рассматривать эту методологию исследований коммуникационных протоколов, а сконцентрируемся на методах, имеющих экспериментальный характер.

Создание и анализ аналитических моделей коммуникационных протоколов является непростой задачей, поэтому более распространенным методом исследований является имитационное, а также натурное моделирование. При использовании этого подхода имитационная модель становится источником исходных данных. Такими данными могут быть трассы о зарегистрированных событиях, произошедших в коммуникационной сети, а также другие статистические данные о поведении исследуемого протокола.

Подходы, основанные на использовании имитационных моделей, могут использовать только оригинальное программное обеспечение. В этом случае исследователь самостоятельно, с нуля, создает программные модели протоколов, узлов коммуникационной сети и каналов связи. На основе этих моделей ставятся эксперименты, получаются исходные данные, которые потом анализируются и представляются автором в виде научного результата. Примером использования такой методологии исследований является исследование протокола ARTCP (Adaptive Rate TCP) [2]. Очевидны недостатки такого подхода: трудно полагаться на такие модели, так как они редко проходят всестороннюю экспертизу научного сообщества, могут содержать ошибки, неточности, быть недоступными из-за лицензионных ограничений. Кроме того, модель может быть потеряна, может быть прекращена ее поддержка в связи с изменением научных интересов автора. Такие модели являются трудоемкими в разработке, так как для всесторонних исследований сравнительного характера необходимо реализовывать версии нескольких коммуникационных протоколов и сред передачи. Оригинальный характер модели и присущие ей недостатки могут оказывать негативное влияние на качество исходных данных, соответственно, как бы далее не производился анализ, результаты будут недостоверными. Такой подход к анализу коммуникационных протоколов являлся популярным, когда были недостаточно развиты пакеты для моделирования коммуникационных сетей.

Пакеты для имитационного моделирования коммуникационных сетей создавались для

преодоления недостатков оригинальных моделей. Примерами таких пакетов являются системы ns-3 [4], [11] (и ее предшественница, система ns-2 [3]), OMNeT++ [5], Cisco Packet Tracer [6] и многие другие. Квалифицированный коллектив авторов таких средств позволяет создавать качественный переносимый код, доступный для экспертизы широкого круга ученых (естественно, при условии соответствующего способа лицензирования). Если исходный код пакета моделирования доступен, как, например, в случае с ns-3 и OMNeT++, то вокруг такого проекта возникает сообщество, что тоже позволяет быстро находить и исправлять ошибки, тем самым повышая качество моделирования. Средства моделирования ns-3 и OMNeT++ по сути представляют собой библиотеки для разработчиков имитационных моделей сетей, начиная от самых простых до весьма нетривиальных, включая модели коммуникационных протоколов, от канального до уровня приложений. Независимые исследователи могут разрабатывать модели своих, оригинальных протоколов, проводить их анализ как со стандартными аналогичными протоколами, так и с другими, код которых является открытым. Наличие широко признанных средств имитационного моделирования позволяет получать более надежные исходные данные для анализа и более того, позволяет решить проблему получения этих данных независимыми исследовательскими группами. Однако при проведении исследований при помощи такого подхода необходимо помнить, что анализируется лишь модель протокола, а то, как протокол будет вести себя в реальных условиях может отличаться от поведения модели [1].

Соответственно, чтобы решить эту задачу, можно реализовать исследуемый протокол непосредственно в ядре операционной системы и получить исходные данные, приближенные к реальности. Однако такой подход имеет несколько недостатков. Реализация протокола в ядре операционной системы является сложной задачей из области системного программирования, требующей учета архитектурной специфики операционной системы. Исходный код такой операционной системы должен быть открыт, поэтому выбор чаще всего останавливается на системе Linux. Ядро этой операционной системы не является хорошо документированным ПО, поэтому свой протокол реализовать труднее, чем в пакете для имитационного моделирования. Отладка и поиск ошибок в разработанном модуле ядра также является сложным занятием. Кроме того, даже в случае успешной реализации, исследователь ограничен теми сетевыми топологиями, которые могут быть физически им созданы. Еще одним ограничением может являться отсутствие адекватных реализаций протоколов, традиционно считающихся эталонными с точки зрения анализа, т.е. мы ограничены только теми реализациями, которые есть в данной операционной системе. Другим недостатком такого подхода является то, что реализация по сравнению с моделью перегружена деталями, ее трудно понять другому ученому, следовательно экспертная оценка ее корректности и соответствия опубликованному

алгоритму работы протокола существенно затруднена. Тем не менее, успешные попытки применения такого подхода предпринимаются (см. [17], [19], [23]).

Существует также несколько комбинированных подходов, призванных сгладить недостатки подходов, основанных на использовании только систем имитационного моделирования, либо только реализаций протоколов. Например, в системе ns-3 реализован модуль DCE (Direct Code Execution) [21], который позволяет запускать в среде ns-3 реализации протоколов, выполняющихся в пространстве ядра, либо пользовательском пространстве. Необходимо подчеркнуть, что эта возможность предназначена для повышения уровня реализма, чтобы таким образом, приблизить получающиеся на модели исходные данные к тем, которые наблюдаются в реальности. Кроме того, в ns-3 возможна передача данных не только с использованием существующих моделей элементов коммуникационной архитектуры, но и организация взаимодействия с использованием реальных устройств и каналов связи. С другой стороны, ограничение с точки зрения инфраструктуры, существующее для реальных реализаций протоколов, может преодолеваться с помощью эмуляции сетевой топологии при помощи технологий виртуализации [1] узлов коммуникационной сети, сохраняя реализм исполнения протоколов на этих узлах.

Таким образом, в настоящее время существует широкий набор средств имитационного и натурального моделирования коммуникационных протоколов, среди которых ученые могут выбирать, руководствуясь необходимым уровнем простоты разработки и реализма поведения.

IV. Критика существующих методологий проведения экспериментальных исследований

Если мы посмотрим на рис. 1, и попробуем его сопоставить с рассмотренными в предыдущем разделе методологиями, то можно увидеть, что они затрагивают только процесс получения исходных данных о работе коммуникационной системы, и, возможно, получение аналитических данных.

Действительно, при использовании первого рассмотренного подхода, оригинальных систем имитационного моделирования, может возникать сомнение в полученных результатах моделирования, так как научная экспертиза оригинальной системы могла не являться всесторонней. Решением этой проблемы является использование систем, которые являются стандартом де-факто в научном мире. К таким системам можно отнести, например, систему ns-3 (около пяти тысяч упоминаний в Google Scholar, а на официальном сайте [4] можно найти ссылки на сотню научных публикаций) или OMNeT++ (около трех тысяч упоминаний в Google Scholar, на официальном сайте можно найти ссылки более чем на 450 публикаций). Наличие надежной системы моделирования позволяет лишь получить более достоверные исходные данные о работе исследуемых протоколов.

Использование реализаций позволяет увидеть, как протокол работает в реальных условиях, однако в этом случае для стороннего исследователя становится трудно сопоставить реализацию и алгоритм, который представляет автор протокола. Такая ситуация характерна, например, в случае протокола Trickle [17]. В этом примере автор предоставил доступ к исходным кодам, реализующим протокол в ОС Linux, а также в статье [17] описал принцип его работы. В результате реализация оказалась перегружена деталями и это вряд ли делает возможным понять принципы ее работы кому-то кроме самого автора. Осложняет ситуацию отсутствие документации к программному коду. Все это затрудняет воспроизведение результатов исследований и является, по-видимому, неотъемлемым недостатком низкоуровневых реализаций коммуникационных протоколов. Имитационные модели в этом плане смотрятся выигрышнее, так как они не перегружены деталями реализации, могут быть документированы (например, сама система ns-3 и встроенные модели документированы при помощи Doxygen).

В контексте компромисса между моделью протокола и его низкоуровневой реализацией необходимо отметить исследование [19], в рамках которого была создана и модель протокола в системе ns-2 [3] и его реализация для ОС Linux (доступны по адресу [20]). В рамках такой методологии можно получить не только исходные данные на модели, которые позволяют обнаружить наиболее существенные особенности протокола, но и данные на реализации и это позволяет проанализировать то, как на функционирование протокола могут влиять детали реализации. Этот подход, от модели к реализации, представляется нам методологически корректным и позволяет приблизиться к получению достоверных исходных данных. Необходимо отметить, что в области коммуникационных протоколов сложилась практика публикации исходных кодов своих моделей и реализаций протоколов в сети Интернет. Это говорит о понимании того, что алгоритм протокола, описанный в статье, вместе с анализом производительности этого протокола не являются достаточными условием для эффективного обмена научными знаниями и для формирования адекватной оценки корректности научных выводов. Иногда даже оригинальные протоколы включаются в стандартную поставку средств имитационного моделирования, например, как это было с протоколом TCP Westwood (и его последователя TCP Westwood+) [22], которые сейчас входят в состав системы ns-3 (на сайте [23] можно найти также исходные коды моделей для системы ns-2 и реализацию TCP Westwood в ОС Linux).

Несмотря на успехи и стремление к проведению воспроизводимых исследований, которое выражается в развитии средств имитационного моделирования и публикации исходных кодов протоколов, до проведения полностью воспроизводимых исследований еще далеко. Действительно, если посмотреть на рис. 1, то можно понять, что сейчас читатель обладает в лучшем случае средствами для генерирования исходных данных, а

также собственно текстом научной статьи, однако все, что находится между этими двумя объектами, ему недоступно.

V. Необходимые элементы воспроизводимого исследования

Итак, на сегодняшний момент ситуация складывается таким образом, что в случае оценки научного результата в области коммуникационных транспортных протоколов мы в лучшем случае имеем статью, представляющую результат, и некий исходный код, который является моделью протокола в одной из доступных систем моделирования, либо реализацию протокола в одной из операционных систем. Фактически, полноценно мы имеем статью, самый правый элемент с рис. 1 и отчасти средство, которое может сгенерировать исходные данные.

Модель или реализация протокола сами по себе не могут генерировать исходные данные для анализа производительности протокола, в любом случае необходимо повторно, действуя указаниям авторов в статье, повторять их эксперимент. Здесь можно столкнуться с такими подводными камнями, как например неполное описание эксперимента, что делает исследование невозможным.

Таким образом в части генерирования исходных данных воспроизводимое исследование должно обладать следующими элементами:

- Доступный документированный исходный код модели/реализации протокола. Это необходимо для того, чтобы убедиться, что данная программа адекватно реализует алгоритм работы протокола. Документация добавляет смысла исходному коду и позволяет сделать его намного более читаемым.
- Доступный документированный исходный код для запуска отдельных экспериментов. Читатель должен иметь возможность провести эксперимент в том виде, в каком его проводил исследователь. Документация позволяет разобраться в сути эксперимента и в том, как использовать этот код.
- Наборы исходных данных. В том случае, если исходный код модели/протокола/эксперимента не являются доступными из-за каких-то ограничений, необходимо предоставить как можно более полные наборы исходных данных эксперимента — трассы зарегистрированных событий в процессе эксперимента или иных данных. Наличие исходных данных полезно в любом случае, так как даже при наличии, скажем, исходного кода модели, она может перестать быть работоспособной по разным причинам, не зависящим от автора. Например, значительная часть исследований, проведенных в 2000-х годах, использовала систему ns-2, тогда как сейчас более современной является система ns-3, которая не обладает обратной

совместимостью. Аналогичные рассуждения справедливы и для реализаций протоколов. Так, ряд очень интересных реализаций был проведен для версий ОС Linux, которые сейчас не являются актуальными. Необходимо отметить, что это требование может быть трудно выполнимым, так как исходные данные могут быть весьма объемными.

Для проведения остальных стадий воспроизводимого исследования в области коммуникационных протоколов нет сложившейся практики, и здесь подходящими для проведения воспроизводимых исследований могут оказаться подходы, разработанные в области науки и данных (data science) [14]. Действительно, после того как исходные данные получены с использованием системы имитационного моделирования, либо реализации протокола, то дальнейший их анализ по своему характеру не отличается от анализа любых других статистических данных. Можно считать, что исходными данными являются трассы, которые состоят из зарегистрированных событий (например, получения или отправки пакета данных узлом сети), произошедших во время эксперимента. Сейчас для анализа таких данных в рамках каждого исследования разрабатываются оригинальные средства, которые позволяют строить научные выводы, а на их основе графики и другие формы представления выводов. Однако нам представляется более подходящим использовать специализированные статистические пакеты, такие как R [12], [13]. Далее мы будем рассматривать возможности только пакета R, который является свободным программным обеспечением, предназначенным для проведения статистических вычислений.

Соответственно, процесс проведения исследования должен быть дополнен следующими этапами:

- Код, который читает исходные данные и производит их преобразование в аналитические данные. Этот код должен быть доступен и документирован. Аналитические данные должны быть доступны, особенно в том случае, если исходные данные являются объемными и не могут быть предоставлены в полной мере. В реальности исходные данные представлены в одном из форматов, соответствующих использованному пакету имитационного моделирования. На практике также часто используемыми являются pcap-файлы, которые могут хранить историю пакетов, зафиксированную непосредственно с сетевых интерфейсов. Поэтому возможно написание стандартной библиотеки для R, которая бы облегчала получение аналитических данных. Нами в настоящее время ведется разработка такой библиотеки.
- Доступный документированный исходный код для обработки аналитических данных. Этот код является оригинальным для каждого

исследования, а в результате его выполнения генерируются вычислительные результаты, являющиеся научными выводами исследования.

- Доступный исходный код для визуализации вычислительных результатов, которые впоследствии ложатся в основу научной статьи.

Вернемся теперь к идее грамотного программирования (literate programming), предложенной Д. Кнудом. Использование этой идеи является плодотворным в контексте предложенного подхода, где исходные данные генерируются системой имитационного моделирования или реализацией протокола, а статистический анализ и визуализация результатов этого анализа происходят в системе R. Результат такого программирования должен обладать следующими свойствами: он должен содержать текст на естественном языке, который объясняет смысл проводимого анализа, и программный код, который производит необходимые вычисления. На основе такого комбинированного документа должен генерироваться отчет, который может быть прочитан ученым. В системе R имеется два пакета, которые обладают такими возможностями. Пакет Sweave [28] использует для документирования язык LaTeX, куда могут быть внедрены выражения языка R. Документ Sweave во время компиляции производит вычисление R-выражений и компилирует файл в системе TeX. Другим является пакет knitr [29], который в качестве языка документирования может использовать несколько языков — LaTeX, HTML или Markdown, а в качестве языка программирования так же как и Sweave использует R. Таким образом с использованием этих средств может быть сформирован документ, объединяющий и систему документирования, и язык программирования, позволяющий проводить научные вычисления.

Осталось рассмотреть последнюю задачу, а именно каким образом можно распространять данные и сопутствующий код. На первый взгляд это может быть затруднительным, так как не каждый ученый способен организовать веб-сайт для представления результатов исследований и обеспечить его бесперебойную работу в течение длительного периода времени, пока его работа остается актуальной. Однако для этого можно использовать существующие репозитории кода, такие как GitHub [26] и Bitbucket [27]. Таким образом, в статье, которая содержит воспроизводимый результат, должен быть указан соответствующая ссылка на репозиторий.

Таким образом, в настоящее время создана вся инфраструктура и все возможности для проведения воспроизводимых научных исследований в области коммуникационных систем.

VI. Заключение

В статье проведен анализ современных подходов, использующихся для проведения анализа производительности коммуникационных протоколов,

и показано, что процесс проведения исследований в этой области требует усовершенствований чтобы представляемые результаты исследований обладали свойством воспроизводимости. Это качество стало возможным, благодаря наличию средств имитационного моделирования, являющихся стандартами де-факто в рассматриваемой области исследований, а также благодаря применению методов исследований, аналогичных тем, что используются в области наук о данных.

Благодарности

Работа поддержана грантом РФФИ, проект 14-01-31539 "Моделирование, анализ и верификация новых транспортных протоколов для современных коммуникационных сервисов".

References

- [1] Handigol N., Heller B., Jeyakumar V., Lantz B., McKeown N. Reproducible Network Experiments Using Container-Based Emulation // Proc. of ACM CoNEXT'12, France, December 10–13, 2012. p. 253–264
- [2] Alekseev I., Sokolov V. ARTCP: Efficient Algorithm for Transport Protocol for Packet Switched Networks // Proc. of Parallel Computing Technologies (PaCT-2001), Novosibirsk, Russia, September 3–7, 2001. Springer. p. 159–174.
- [3] ns-2 official web-site: <http://nsnam.isi.edu/nsnam/index.php>
- [4] ns-3 official web-site: <http://www.nsnam.org/>
- [5] OMNeT++ official web site: <http://www.omnetpp.org/>
- [6] Cisco Packet Tracer web site: <http://www.netacad.com/ru/web/about-us/cisco-packet-tracer>
- [7] Knuth D.E. Literate Programming. // The Computer Journal [Online], 27:97–111, 1984.
- [8] Claerbout J. Electronic Documents Give Reproducible Research a New Meaning // Proc. 62nd Ann. Int. Meeting of the Soc. of Exploration Geophysics, pp. 601–604, 1992.
- [9] Buckheit J.B., Donoho D.L. Wavelab and reproducible research // Time, 474:55–81, 1995.
- [10] Vandewalle P., Kovacevic J., Vetterli M. Reproducible Research // Computing in Science Engineering, pp. 5–7, 2008.
- [11] Henderson T.R., Lacage M., Riley G.F. Network Simulations with the ns-3 Simulator // Proc. of ACM SIGCOMM'08, August 17–22, 2008. p. 527.
- [12] Kabacoff R.I. R in Action // Manning Publications. 2011. 474 p.
- [13] The R Project for Statistical Computing. web site: <http://www.r-project.org>.
- [14] Peng R.D. Tutorial: Methods for Reproducible Research // ENAR'2009. March, 16. Texas, USA.
- [15] Doxygen. www.doxygen.org.
- [16] Willinger W., Taqqu M.S., Leland W.E., Wilson D.V. Self-Similarity in High-Speed Packet Traffic: Analysis and Modeling of Ethernet Traffic Measurements // Statistical Science Vol. 10, Number 1, 1995. pp. 67–85.
- [17] Shieh A., Myers A.C., Sizer E.G. A Stateless Approach to Connection-Oriented Protocols // ACM Transactions on Computer Systems, Vol. 26, No 3, Article 8. 2008. 50 p.
- [18] Avigad J., Harrison J. Formally Verified Mathematics // Communications of the ACM, Vol. 57, No 4. 2014. pp. 66–75.
- [19] Alizadeh M., Greenberg A., Maltz D.A., Padhye J., Patel P., Prabhakar B., Sengupta S., Sridharan M. Data Center TCP (DCTCP) // Proc. of SIGCOMM'2010, August 30–September 3, 2010. New Delhi, India. 12 pp.
- [20] Data Center TCP web page: <http://simula.stanford.edu/alizade/Site/DCTCP.html>

- [21] Tazaki H., Urbani F., Mancini M., Lacage M., Camara D., Turletti T., Dabbous W. Direct Code Execution: Revisiting Library OS Architecture for Reproducible Network Experiments // Proc. of CoNEXT'2013, Santa Barbara, California, USA, December 9–12, 2013. pp. 217–228.
- [22] Grieco L.A., Mascolo S. Performance Evaluation and Comparison of Westwood+, New Reno and Vegas Congestion Control // ACM Computer Communication Review, Vol. 34(2), April, 2004.
- [23] C3Lab Westwood site: <http://c3lab.poliba.it/index.php/Westwood>
- [24] Никитинский М., Чалый Д.Ю. Сравнительный анализ производительности транспортных протоколов Trickle и TCP в условиях высокой нагрузки на коммуникационную сеть // Моделирование и анализ информационных систем, Т. 19, No 4, 2012. с. 37–47.
- [25] Paxson V. Measurements and Analysis of End-to-End Internet Dynamics // PhD Thesis, University of California, Berkeley, 1997. 409 p.
- [26] GitHub official web site: www.github.com
- [27] Bitbucket official web site: www.bitbucket.org
- [28] Sweave web site: <http://www.statistik.lmu.de/~leisch/Sweave>
- [29] knitr web site: <http://yihui.name/knitr/>