

Анализ производительности сетей 5G миллиметрового диапазона при передаче TCP-данных¹

А.Н. Красилов*, М.В. Суслопаров**, О.О. Филатов**, Е.М. Хоров*

*Институт проблем передачи информации им. А.А. Харкевича РАН, Москва, Россия

**Национальный исследовательский университет «Высшая школа экономики», Москва, Россия

Поступила в редколлегию 05.03.2019 г.

Аннотация—Для обеспечения высоких скоростей передачи данных в сетях 5G планируется использовать миллиметровый диапазон частот. Однако производительность таких сетей существенно снижается из-за протокола TCP, используемого различными приложениями для передачи данных. В работе показано, что из-за алгоритма медленного старта, используемого в протоколе TCP, скорость загрузки данных в сетях 5G оказывается значительно ниже доступной пропускной способности беспроводного канала. Проведен анализ различных подходов к решению данной проблемы и предложено использование платформы xStream для выбора параметров протокола TCP в сетях 5G. Данный подход позволяет значительно повысить скорость загрузки данных и при этом не требует изменения программного обеспечения пользовательских устройств, что обеспечивает возможность быстрого внедрения данного решения на практике.

КЛЮЧЕВЫЕ СЛОВА: сети пятого поколения, миллиметровый диапазон частот, TCP, облачные сервисы, скорость доставки данных.

1. ВВЕДЕНИЕ

Одним из ключевых требований к беспроводным сетям пятого поколения (5G) является обеспечение высоких скоростей передачи данных: средняя скорость передачи данных на пользователя должна быть более 100 Мбит/с, а пиковая скорость на соту достигать 20 Гбит/с [1]. Такие скорости передачи данных необходимы для поддержки новых приложений и сервисов [2], таких как передача трехмерных видеоизображений и видеоизображений сверхвысокой четкости, обеспечение мгновенной доставки больших объемов данных с облачных сервисов на устройство пользователя (англ. instant cloud access) и др.

Согласно известным положениям теории информации, для достижения таких высоких скоростей передачи данных необходимы широкие частотные каналы. Существующие диапазоны частот, используемые для мобильной радиосвязи (несущая частота <6 ГГц), содержат частотные каналы шириной порядка 100 МГц. В сетях 5G помимо традиционных диапазонов частот планируется использовать миллиметровый диапазон (англ. mmWave) на несущих частотах от 24 до 86 ГГц, в которых доступны частотные каналы шириной до 6 ГГц.

Использование миллиметрового диапазона частот с широкими частотными каналами с одной стороны позволяет существенно повысить потенциально достижимую скорость передачи данных. С другой стороны, передача сигналов в миллиметровом диапазоне частот подвержена значительному ослаблению с расстоянием, а также существенной флуктуации мощности принимаемого сигнала со временем. В связи с этим, во-первых, значительно снижается радиус

¹ Исследование выполнено в ИППИ РАН за счет гранта Правительства Российской Федерации (Договор No 14.W03.31.0019)

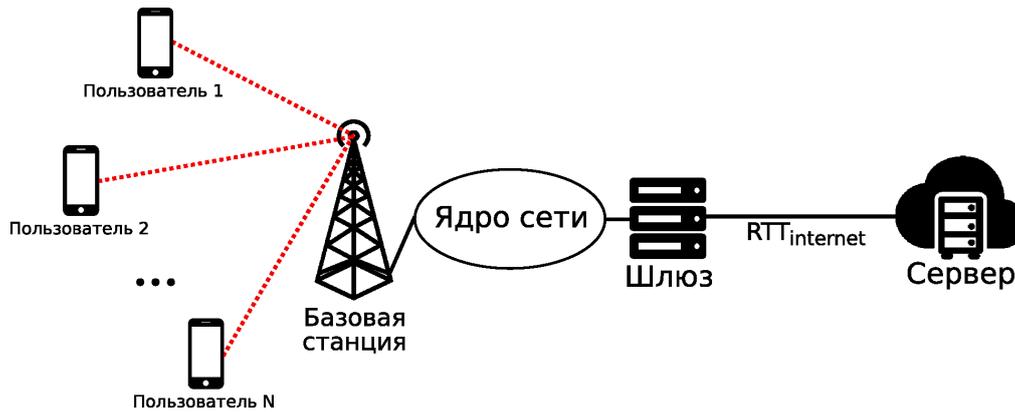


Рис. 1. Рассматриваемый сценарий.

покрытия сети (до сотен метров), а, во-вторых, скорость передачи данных может существенно меняться со временем: даже при отсутствии движения пользователя – в диапазоне от десятков Мбит/с до нескольких Гбит/с [3].

В данной работе рассматривается сценарий передачи данных (файлов) от сервера, расположенного в облаке, на устройства пользователей, подключенных к сети 5G mmWave (см. рис. 1). Для обеспечения надежной и упорядоченной доставки данных на сервере и пользователях используется протокол транспортного уровня TCP (Transmission Control Protocol). В работе показано, что из-за алгоритма медленного старта (англ. Slow Start), используемого в протоколе TCP, скорость загрузки данных в сетях 5G mmWave оказывается значительно ниже доступной пропускной способности беспроводного канала. В работе исследованы различные подходы к решению данной проблемы. Показано, что использование платформы xStream [4, 5], обеспечивающей передачу контрольной информации между устройствами беспроводной сети и сервером, позволяет существенно повысить производительность сетей 5G mmWave при передаче TCP-данных, которая становится близкой к теоретически достижимой верхней границе.

Дальнейшее изложение работы построено следующим образом. В разделе 2 анализируется проблема медленного старта. В разделе 3 рассмотрены различные подходы к решению данной проблемы. В разделе 4 проведено исследование данных подходов с помощью среды имитационного моделирования Network Simulator 3 (NS-3) [6]. В разделе 5 резюмируются основные результаты работы.

2. ПРОБЛЕМА МЕДЛЕННОГО СТАРТА TCP

Рассмотрим процесс передачи данных от сервера к пользователю при использовании протокола TCP. Передача состоит из нескольких этапов (см. рис. 2).

1. Установка соединения между клиентом и сервером с помощью процедуры тройного рукопожатия: отправка клиентом запроса на соединение (SYN), отправка сервером ответного запроса и подтверждения (SYN/ACK), отправка клиентом подтверждения соединения (ACK).
2. Передача данных: отправка клиентом запроса на передачу данных, отправка сервером данных (файла). При получении пакетов с данными каждая из сторон соединения отправляет пакеты подтверждения (ACK).
3. Закрытие соединения: отправка клиентом запроса на закрытие соединения (FIN), отправка сервером подтверждения (ACK) и ответного запроса на закрытие соединения (FIN), отправка клиентом подтверждения (ACK).

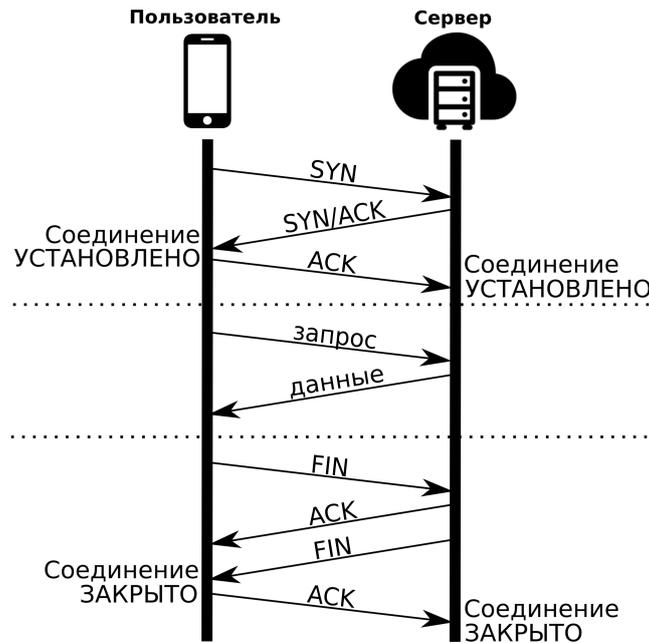


Рис. 2. Передача данных с помощью протокола TCP.

После установления TCP-соединения для передачи данных используется алгоритм медленного старта (англ. Slow Start) [7]. Данный алгоритм позволяет постепенно увеличивать объем данных, отправляемых сервером в единицу времени. Суммарный объем данных, отправляемых сервером без подтверждения, называется окном перегрузки (англ. congestion window, CWND), которое измеряется в байтах. Обозначим за W_0 начальное окно перегрузки на сервере. Окно перегрузки увеличивается на один сегмент, который имеет размер MSS байт, с каждым полученным подтверждением. Из этого следует, что с каждым RTT (Round-Trip Time) – временем между отправкой сегмента и получением подтверждения на него – окно перегрузки возрастает вдвое. Таким образом, размер окна перегрузки с течением времени можно оценить следующим образом:

$$W(t) = W_0 2^{t/RTT}. \quad (1)$$

Для использования всей доступной пропускной способности сети окну перегрузки необходимо достигнуть такого уровня, чтобы скорость отправки пакетов сервером была равна минимальной пропускной способности соединения на пути от сервера к пользователю (пропускной способности «узкого места» сети). Тогда размер окна перегрузки при достижении насыщения сети можно оценить следующим образом:

$$W_C = C \cdot RTT, \quad (2)$$

где C – пропускная способность узкого места сети. В случае беспроводной сети узким местом, как правило, является беспроводное соединение.

Из формул (1) и (2) можно найти количество интервалов RTT , которое требуется для перехода от начального окна перегрузки к значению W_C :

$$N_C = \log_2 \left(\frac{C \cdot RTT}{W_0} \right).$$

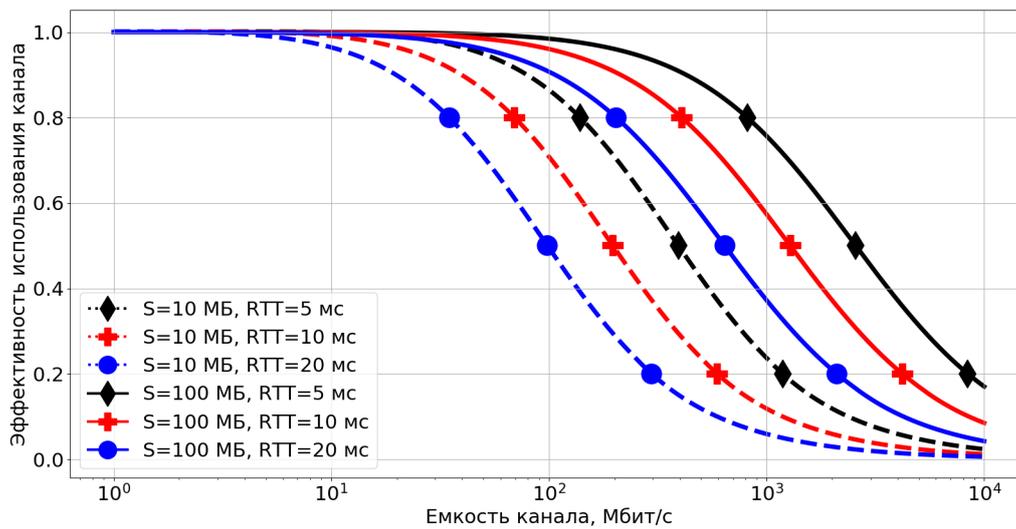


Рис. 3. Эффективность использования беспроводного канала при передаче файла.

Размер данных, передаваемых за N интервалов RTT при использовании процедуры медленного старта, можно оценить следующим образом:

$$S(N) = \sum_{i=0}^N W_0 2^i = W_0(2^{N+1} - 1).$$

Отсюда можно оценить время, которое потребуется для того, чтобы загрузить файл размера S при использовании процедуры медленного старта:

$$\tau(S) = \left(\log_2 \left(\frac{S}{W_0} + 1 \right) - 1 \right) RTT.$$

Учитывая, что после достижения окна перегрузки значения W_C скорость загрузки данных становится ограниченной пропускной способностью беспроводного соединения C , общее время загрузки файла можно оценить следующим образом:

$$T(S) = \begin{cases} \tau(S), & S \leq S(N_C) \\ N_C RTT + \frac{S - S(N_C)}{C}, & S > S(N_C) \end{cases}$$

Определим величину эффективности использования ресурсов беспроводного канала при передаче файла размера S как:

$$\xi = \frac{S/T(S)}{C}.$$

Исследуем зависимость ξ от пропускной способности беспроводного канала C , времени круговой передачи пакета RTT и размера загружаемого файла S . Из рис. 3 можно видеть, что при малых значениях пропускной способности беспроводного канала (1–10 Мбит/с) алгоритм медленного старта не оказывает значительного влияния на эффективность использования канала. При больших значениях пропускной способности канала (1–10 Гбит/с) эффективность использования канала значительно снижается. Особенно ярко этот эффект проявляется при загрузке файлов небольшого размера (10 МБ), так как при таких размерах данных алгоритм медленного старта не достигает значения W_C до окончания загрузки файла.

Из проведённого анализа можно сделать вывод, что при использовании беспроводных соединений с пропускной способностью более гигабита в секунду и небольшими потоками данных необходимо модифицировать алгоритм медленного старта для повышения эффективности использования ресурсов беспроводного канала.

3. МЕТОДЫ РЕШЕНИЯ ПРОБЛЕМЫ МЕДЛЕННОГО СТАРТА

Рассмотрим существующие методы решения проблемы медленного старта.

Одним из методов является установка прокси-сервера (англ. performance-enhancing proxy) на промежуточных узлах сети [8]. Данный метод позволяет разделить TCP-соединение между сервером и пользователем на два или более сегмента. Как видно из формулы (2), размер окна перегрузки для достижения насыщения сети прямо пропорционален значению RTT . При использовании прокси-сервера RTT на каждом TCP-соединении уменьшается, и, как следствие, уменьшается время, за которое процедура медленного старта увеличит окно перегрузки до достижения насыщения сети.

Другим методом решения проблемы медленного старта является установка начального окна перегрузки в значение W_C согласно формуле (2). Для этого необходимо знать значение круговой задержки передачи пакета RTT и пропускной способности C узкого места сети. RTT на сервере можно оценить, измерив разницу между временем отправки пакета SYN/ACK и временем получения подтверждения ACK от пользователя (см. рис. 2).

Одним из способов нахождения пропускной способности узкого места сети является использование алгоритма Quick-Start [9]. Согласно алгоритму Quick-Start, сервер записывает в IP-заголовок пакета SYN/ACK «желаемую» скорость отправки данных. При получении данного пакета промежуточным устройством (например, маршрутизатором) оно, в случае поддержки Quick-Start, может подтвердить, уменьшить или отклонить запрашиваемую скорость передачи данных. Информация, содержащая значение запрошенной скорости (которое, возможно, было изменено промежуточными устройствами), пересылается пользователем назад в пакете ACK. На основе полученной информации из пакета ACK сервер может оценить максимально возможную скорость отправки данных по сети для выбора подходящего окна перегрузки согласно формуле (2). Чтобы избежать переполнения очередей на промежуточных сетевых узлах при отправке большого объёма данных после установления TCP-соединения, применяется алгоритм TCP Raising [10]. При работе данного алгоритма пакеты отправляются сервером с задержкой, соответствующей выбранной скорости передачи данных. Информация о пропускной способности сети, полученная сервером в процессе использования алгоритма Quick-Start, позволит корректно настроить параметры задержки для алгоритма TCP Raising.

Основной недостаток Quick-Start заключается в том, что для его реализации необходимы модификации стека протоколов TCP/IP на всех устройствах сети, в том числе на стороне пользователя. Это, в свою очередь, требует внесения изменений в операционную систему (ОС). Одним из методов решения проблемы медленного старта, который не требует изменения ОС пользователя, является использование платформы xStream [4]. Данная платформа позволяет устанавливать дополнительные TCP-соединения между сетевыми устройствами (например, базовой станцией) и оконечными устройствами (например, сервером) с целью передачи служебной информации, такой как размер передаваемых данных и пропускная способность сети. Таким образом, получив с помощью платформы xStream оценку пропускной способности беспроводного канала C и оценив RTT , сервер может сразу после установления TCP-соединения с пользователем установить размер начального окна перегрузки согласно формуле (2). Следует отметить, что для реализации данного подхода требуется изменение программного обеспечения (а именно приложений) только на сервере и базовой станции и не требуется изменение ОС пользователя и других сетевых устройств.

Таблица 1. Основные параметры имитационной модели.

Параметр	Значение
Модель распространения сигнала	3GPP mmWave
Ширина полосы	1 ГГц
Несущая частота	26 ГГц
Мощность БС	30 дБм
Высота gNB	10 м
Высота UE	1,6 м
Алгоритм управления TCP	CUBIC [11]
Максимальный размер сегмента (MSS)	1460 байт
Начальное окно перегрузки	10 MSS
Размер очереди на промежуточных узлах	1000 пакетов
RTT на проводном соединении	10 мс
Пропускная способность проводного соединения	10 Гбит/с
Время одного эксперимента	100 с
Количество независимых прогонов	30

4. ЧИСЛЕННЫЕ РЕЗУЛЬТАТЫ

Исследуем решения, описанные в разделе 3, с помощью среды имитационного моделирования NS-3 [6] с модулем mmWave, описанным в работе [3], и моделью приложения, разработанной в работе [12]. Рассмотрим следующие модификации TCP.

1. TCP-Default – на оконечных устройствах используется протокол TCP с параметрами по умолчанию ($W_0 = 10$ MSS).
2. TCP-Inf-Buff – на оконечных устройствах используется протокол TCP с большим начальным окном перегрузки 10^6 MSS и размером очереди на промежуточных узлах 10^6 пакетов. Такая модификация позволяет анализировать случай, когда после установки TCP-соединения весь файл помещается в очередь на базовую станцию, и, таким образом, ресурсы беспроводного канала используются полностью.
3. TCP-Прoxy – на оконечных устройствах используется протокол TCP с параметрами по умолчанию, а на шлюзе установлен прокси-сервер (см. раздел 3).
4. TCP-CWND – используется платформа xStream, т.е. начальное значение окна перегрузки на сервере устанавливается в соответствии с формулой (2).
5. TCP-CWND-Pacing – используется платформа xStream и алгоритм TCP Pacing [10].

Основные параметры имитационной модели указаны в таблице 1. В экспериментах, представленных ниже, исследуется зависимость скорости загрузки файлов (отношения размера файла к времени его загрузки) от размера файла и количества пользователей, подключенных к сети.

4.1. Влияние размера файла

В первом эксперименте рассматривается один пользователь, расположенный на расстоянии 30 м от базовой станции, который загружает файлы различного размера.

Из рис. 4 можно видеть, что с увеличением размера файла скорость загрузки увеличивается, так как доля времени, используемая на процедуру медленного старта, уменьшается по сравнению с полным временем загрузки файла. Модификация TCP-CWND показывает меньшую производительность, чем TCP-Default, так как в отсутствие процедуры TCP Pacing после

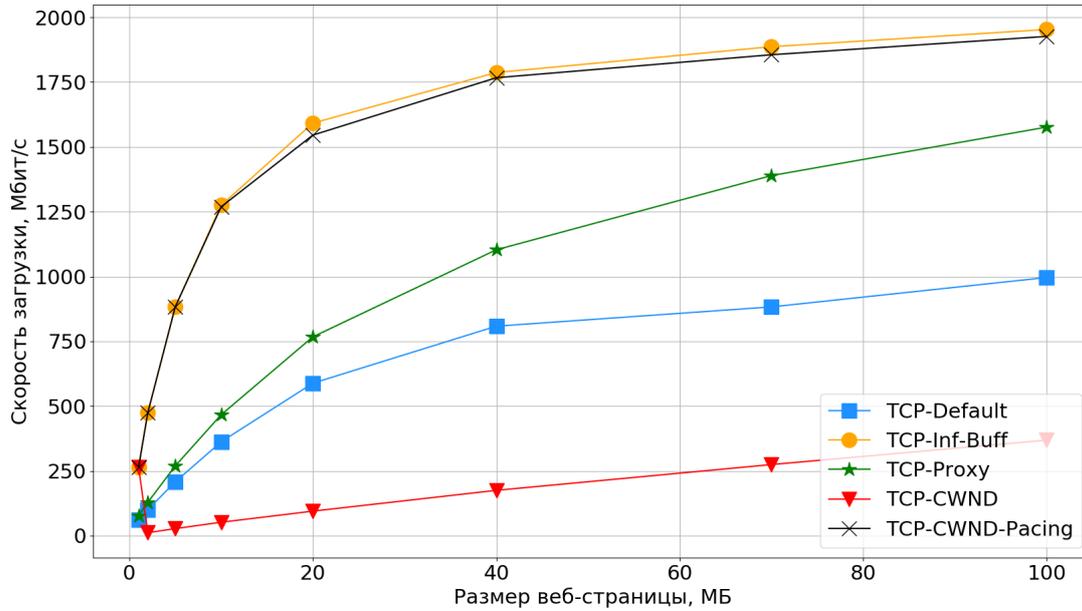


Рис. 4. Зависимость скорости загрузки от размера файла.

установления TCP-соединения отправляется большой объем данных, что приводит к потере пакетов на промежуточных узлах. Поэтому в дальнейших экспериментах данная модификация не рассматривается.

Сравнивая кривые TCP-Proxy и TCP-Default, можно видеть, что TCP-Proxy позволяет увеличить скорость загрузки файла на 30–50%. TCP-Inf-Buff обеспечивает наибольшую скорость загрузки данных за счет того, что файл мгновенно отправляется с сервера на базовую станцию. Производительность TCP-CWND-Pacing близка к верхней границе (кривой TCP-Inf-Buff) и дает прирост производительности от 100 до 300% относительно TCP-Default.

4.2. Влияние числа пользователей

Во втором эксперименте исследуются различные модификации протокола TCP при одновременной работе в сети нескольких мобильных устройств. Пользователи равномерно размещены в круге радиусом $R = 100$ м, в центре которого расположена базовая станция. Размер файла задается усеченным логнормальным распределением со средним значением 30 МБ, со средним квадратическим отклонением 10 МБ, с минимальным и максимальным значением 1 и 100 МБ соответственно. Время между последовательными загрузками файлов на пользователе задается усеченным экспоненциальным распределением со средним значением 5 с, с минимальным и максимальным значением 1 и 10 с соответственно.

Из рис. 5 можно видеть, что с увеличением количества пользователей скорость загрузки уменьшается, так как пропускная способность беспроводного канала распределяется между всеми пользователями. Сравнивая кривые TCP-Default и TCP-Proxy, можно видеть, что TCP-Proxy увеличивает скорость загрузки на 30–40% относительно TCP-Default. Из-за того, что ресурсы распределяются между пользователями в зависимости от состояния очереди на базовой станции, в рассматриваемом сценарии TCP-Inf-Buff не будет являться верхней границей, как в сценарии с одним пользователем. Модификация TCP-CWND-Pacing позволяет увеличить скорость загрузки данных на 75–100% по сравнению с TCP-Default и обеспечивает примерно одинаковую производительность с TCP-Inf-Buff, т.е. со случаем, когда отсутствует

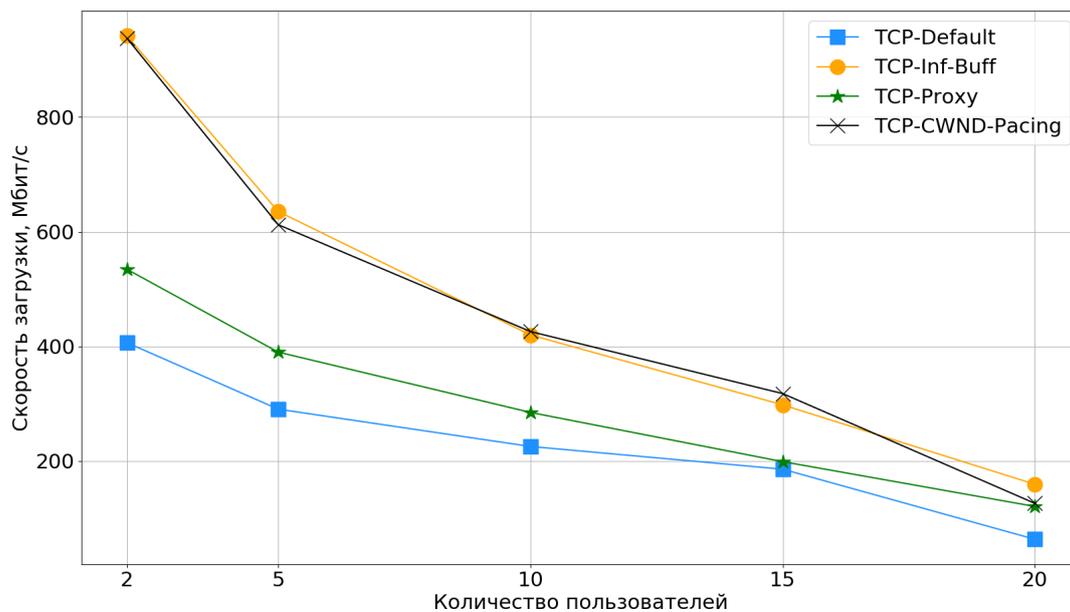


Рис. 5. Зависимость скорости загрузки данных от числа пользователей.

проблема медленного старта. Это означает, что предложенная модификация TCP позволяет решить проблему медленного старта и обеспечивает производительность, близкую к верхней границе.

5. ЗАКЛЮЧЕНИЕ

В данной работе проведен анализ эффективности беспроводных сетей пятого поколения, работающих в миллиметровом диапазоне частот (сетей 5G mmWave), при передаче TCP-данных. Показано, что из-за алгоритма медленного старта, используемого в протоколе TCP, скорость загрузки данных оказывается значительно ниже доступной пропускной способности беспроводного канала. В работе проанализированы различные подходы к решению данной проблемы. С помощью имитационного моделирования в среде NS-3 показано, что подход, основанный на использовании платформы xStream, позволяет существенно повысить производительность сетей 5G mmWave при передаче TCP-данных, которая становится близкой к теоретически достижимой верхней границе. В отличие от других подходов, предложенных в литературе, использование платформы xStream не требует изменения программного обеспечения пользовательских устройств, что обеспечивает возможность быстрого внедрения предложенного решения проблемы медленного старта на практике.

СПИСОК ЛИТЕРАТУРЫ

1. *Framework and Overall Objectives of the Future Development of IMT for 2020 and beyond* // ITU-R, Recommendation M.2083, September 2015.
2. *Service Requirements for Next Generation New Services and Markets* // 3GPP Tech. Rep. TS-22.261, December 2018.
3. M. Mezzavilla et al., *End-to-End Simulation of 5G mmWave Networks* // IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2237-2263, 2018.

4. I. F. Akyildiz, E. Khorov, A. Kiryanov, D. Kovkov, A. Krasilov, M. Liubogoshchev, D. Shmelkin, S. Tang, *xStream: A New Platform Enabling Communication Between Applications and the 5G Network* // IEEE Globecom Workshops (GC Wkshps), December 2018
5. I.F. Akyildiz, A. Kak, E. Khorov, A. Krasilov, A. Kureev, *ARBAT: A flexible network architecture for QoE-aware communications in 5G systems* // Computer Networks, Vol. 147, pp. 262-279, 2018.
6. Network Simulator 3 (NS-3), [Online] Available: <https://www.nsnam.org/>
7. M. Allman, V. Paxson, E. Blanton, *RFC 5681 - TCP Congestion Control*, September 2009, Available: <https://tools.ietf.org/html/rfc5681>
8. J. Border, M. Kojo, J. Griner, G. Montenegro, Z. Shelby, *RFC 3135 - Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations*, June 2001, Available: <https://tools.ietf.org/html/rfc3135>
9. S. Floyd, M. Allman, A. Jain, P. Sarolahti, *RFC 4782 - Quick-Start for TCP and IP*, January 2007, Available: <https://tools.ietf.org/html/rfc4782>
10. A. Aggarwal, S. Savage, T. Anderson, *Understanding the Performance of TCP Pacing* // Proceedings - IEEE INFOCOM 10.1109/INFOCOM.2000.832483, January 2000
11. I. Rhee, L. Xu, S. Ha, A. Zimmermann, L. Eggert, R. Scheffenegger, *RFC 8312 - CUBIC for Fast Long-Distance Networks*, February 2018, Available: <https://tools.ietf.org/html/rfc8312>
12. Белогаев А.А., Жирнов Н.С., Красилов А.Н., Ляхов А.И., Хоров Е.М., *Исследование передачи Web-данных в сотовых сетях* // Информационные процессы, Т. 16, № 4, 2016.

Performance Evaluation of TCP Data Transmission in 5G mmWave Networks

A.N. Krasilov, M.V. Susloparov, O.O. Filatov, E.M. Khorov

To achieve extremely high data rates, 5G systems employ millimeter-wave (mmWave) frequencies. However, the performance of such wireless systems suffers from the Transmission Control Protocol (TCP) widely used today for data transmission. In this paper, we argue that the download rate in 5G networks is significantly lower than the capacity of the mmWave link because of the Slow Start algorithm used in TCP. We analyze various approaches to this problem and propose the usage of the xStream platform for adjusting TCP parameters. This approach significantly increases download rate and does not require any changes to client devices, which simplifies the implementation of the proposed approach in real systems.

KEYWORDS: 5G networks, mmWave frequency band, TCP, cloud services, download rate.