

ПОДХОД К ОПРЕДЕЛЕНИЮ РАЗМЕРОВ БУФЕРА ДАННЫХ ДЛЯ МУЛЬТИМЕДИЙНЫХ ПРИЛОЖЕНИЙ ПРИ ПЕРЕМЕННОЙ ПРОПУСКНОЙ СПОСОБНОСТИ КАНАЛА ПЕРЕДАЧИ ДАННЫХ

Быков Д. В., Зинов П. В., Аверин Е. В.

ГОУ ВПО «Волгоградский государственный технический университет», Волгоград, Россия (400131, г. Волгоград, пр. Ленина, 28), e-mail: guardianzpv@gmail.com

В статье описывается динамический способ определения размера буфера данных в мультимедийном приложении. Для определения теоретической величины размера буфера составлена система из фильтра сетевого трафика, анализатора этого трафика, а также собственного клиент-серверного приложения, написанного на языке высокого уровня Java, позволяющего эмулировать передачу видеопотока от сервера к клиенту. В ходе исследования были получены наилучшие размеры буфера данных при переменной пропускной способности канала передачи данных. Под наилучшими размерами подразумеваются такие размеры буфера данных, которые позволяют быстро загружать часть видеопотока с сервера, и при этом достаточны для того, чтобы не возникало разрывов в воспроизведении видеопотока. На основе анализа полученных данных была составлена формула определения размера буфера, позволяющего воспроизводить видеопоток без перерыва. Произведено дальнейшее исследование этой формулы, которое позволило получить информацию о минимальном времени воспроизведения видеопотока, которого достаточно для дальнейшей аппроксимации размера буфера входных данных с помощью представленной формулы. В заключение был проведен анализ размера буфера данных в приложении YouTube, которое показало, что данное приложение не учитывает изменения в скорости соединения клиента.

Ключевые слова: буфер данных, мультимедийное приложение, видеопоток, пропускная способность, канал передачи данных.

APPROACH TO DETERMINING THE SIZE OF MULTIMEDIA APPLICATIONS DATA BUFFER BASED ON VARIABLE CONNECTION BANDWIDTH

Bykov D. V., Zinov P. V., Averin E. V.

Volgograd State Technical University, Volgograd, Russia (400131, Volgograd, Lenin's avenue, 28), e-mail: guardianzpv@gmail.com

This paper describes a dynamic way to determine the size of the data buffer in the media application. We have created a system to determine theoretical application's buffer size value. This system is composed of the network filter and traffic analyzer, as well as its own client-server application written in high-level language Java, emulating video streaming process from the server to the client. The study received the best size of the data buffer in volatile bandwidth data transfer. Under the best size, we mean the size of the data buffer that allows you to download the video part from the server quickly, and thus is sufficient to avoid any gaps in the playback video. The formula for determining the buffer size was compiled, based on the analysis of the data, received by our experiments. This formula allows buffer to have the exact size for uninterrupted video playback. There was further investigation of this formula to determine minimum video stream playing time, which would be enough for further approximation of the size of the input data buffer. In conclusion, we analyzed YouTube application buffer size. This investigation showed that the application does not account for changes in the speed of the client connection.

Key words: data buffer, multimedia application, video stream, bandwidth, network data channel.

Введение

Ввод и вывод данных играет большую роль в любом приложении. Зачастую он осуществляется средствами операционной системы. При работе с локальными приложениями и стандартным вводом и выводом данных время отклика операционной системы на каждое действие достаточно мало. Однако ситуация меняется в корне при работе с мультимедийными приложениями, работающими по сети, где зачастую передаются

достаточно большие объемы данных при наличии высокого времени отклика между двумя точками взаимодействия. Прежде чем вывести информацию пользователю, приложение должно аккумулировать некоторую ее часть до определенного времени, чтобы не загружать систему частыми вызовами. Для решения этой проблемы используются буферы данных. Однако не всегда большой буфер данных может решить проблему упорядочивания доступа к графической или сетевой подсистеме ОС. Зачастую в мультимедийном сетевом взаимодействии используются различные широко распространенные пакеты передачи подобного рода данных, такие как MPEG. В данной статье мы продемонстрируем то, каким образом должен изменяться размер буфера данных в приложении с передачей видео сигнала по сети в зависимости от скорости соединения.

Собственное исследование

Соответствующие измерения в ходе моделирования процесса передачи видео сигнала производятся с помощью программных продуктов, таких как Dummynet и Wireshark.

Dummynet перехватывает входящие пакеты данных и передает их в систему с помощью специальных каналов. Таким образом, этот инструмент позволяет задавать параметры соединения, такие как скорость передачи данных, процент потери пакетов, задержку и многое другое.

Wireshark – это инструмент для перехвата и анализа протоколов сетевых данных. Wireshark позволяет перехватывать пакеты в режиме реального времени, накладывать определенные фильтры, отображать заголовки пакетов и собирать различную сетевую статистику.

С помощью Dummynet входящий поток информации был сконфигурирован таким образом, чтобы при изменении размера канала передачи данных среднее его значение было фиксировано на 125кбайт/с.

Для эмуляции передачи данных был разработан программный комплекс, состоящий из клиента и сервера, написанных с применением языка высокого уровня Java. Система работает на TCP протоколе, при этом сервер передает видеоданные на клиента, где эти данные перехватываются с помощью Wireshark, с помощью которого анализируется соответствующая статистика пакетов.

Симуляция процесса передачи видеосигнала происходит с помощью клиента, который имеет встроенный буфер данных. При этом симулируется время кодирования/декодирования видеопотока, а также скорость передачи кадров (framerate). Если скорость передачи данных падает ниже, чем framerate видеопотока, то пользователь наблюдает остановку видеосигнала и ожидает, пока буфер не заполнится данными. Здесь мы можем наблюдать важность динамического размера этого буфера. Если он слишком мал, то видеотрансляция будет

постоянно прерываться, а если слишком большим, то пользователь будет должен ждать наполнения буфера дольше необходимого времени, определяемого скоростью передачи данных.



Рис. 1. Красные маркеры сигнализируют о прерывании в воспроизведении видеопотока при изменении ширины пропускной способности канала передачи данных

Одновременно с остановкой видео мы можем наблюдать график загрузки буфера данных:

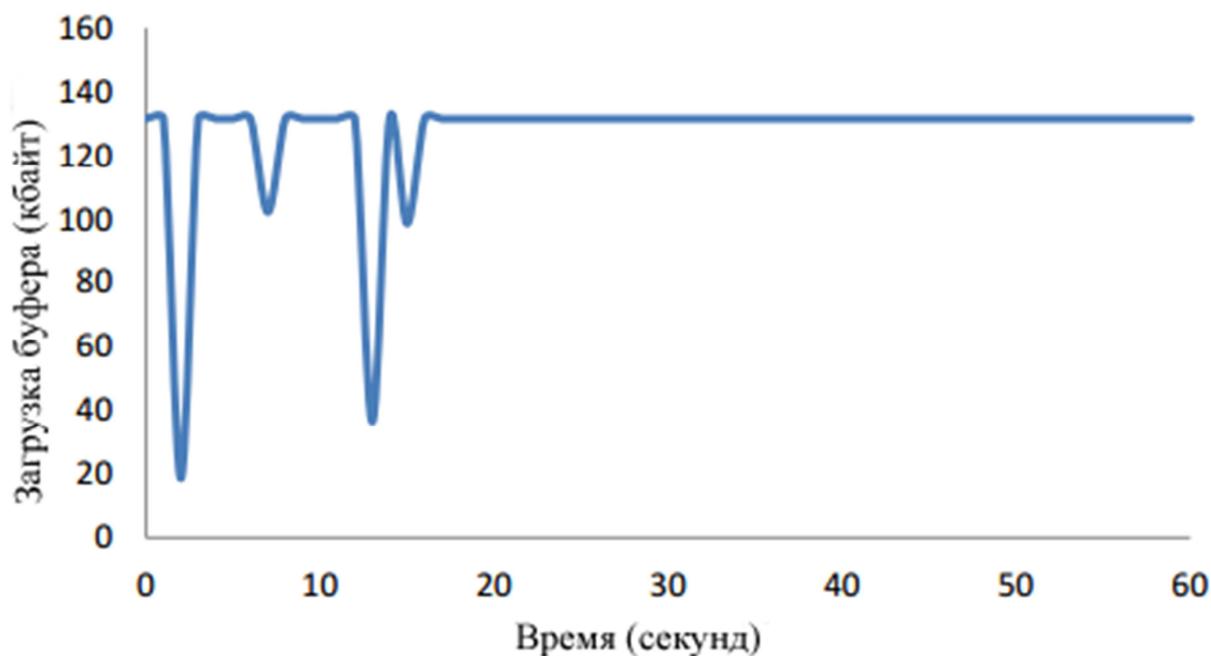


Рис. 2. График загрузки статического буфера данных приложения

Чтобы найти оптимальный размер буфера для текущих параметров скорости соединения и декодирования видео, мы будем добавлять по 1 кбайт к буферу до тех пор, пока прерывания не прекратятся. Эмпирическим путем мы получили значение в 470 килобайт.

Таким образом мы можем задать диапазон изменения скорости передачи данных от -0.5 до +0.5 от оригинального, а затем собрать статистику необходимых размеров буфера для этих параметров. После получения статистики в этом диапазоне мы получаем следующее распределение:

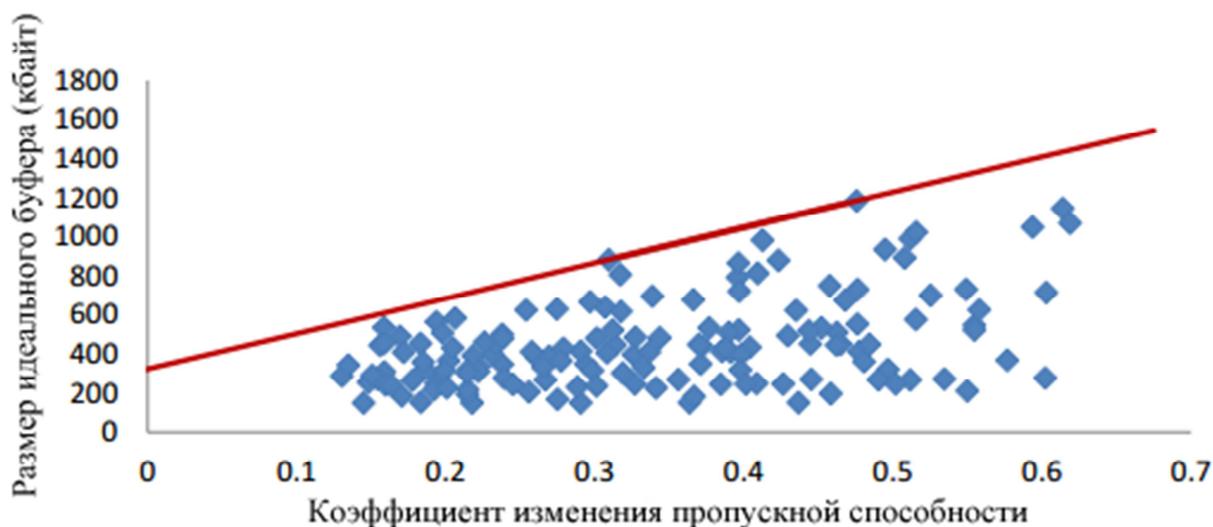


Рис. 3. Распределение размеров статических буферов данных приложения в зависимости от коэффициента изменения пропускной способности канала передачи данных

Получившаяся формула, описывающая размер идеального буфера, при котором не будет происходить прерывов в отображении видео:

$$\text{Предсказанный размер буфера данных приложения} = 1,8 \text{ Мбайт} * \text{КИПС} - 0,32 \text{ Мбайт} \quad (1)$$

Где КИПС – это коэффициент изменения пропускной способности.

До сих пор мы собирали статистику для интервала проигрывания видео в 60 секунд. Проведем подобные измерения для интервала от 2 до 59 секунд с шагом в 1 секунду, а также вычислим предсказанный размер буфера по полученной формуле для каждого значения из этого интервала. В полученных результатах видна корреляция значений в интервале от 0 до 8 секунд.

Для обеспечения корректности сравнения, модифицируем программный комплекс таким образом, чтобы вместо процента потерь пакетов нам была доступна информация о количестве прерываний за интервал. Результат моделирования этой системы в интервале от 0 до 8 секунд показывает следующие результаты:

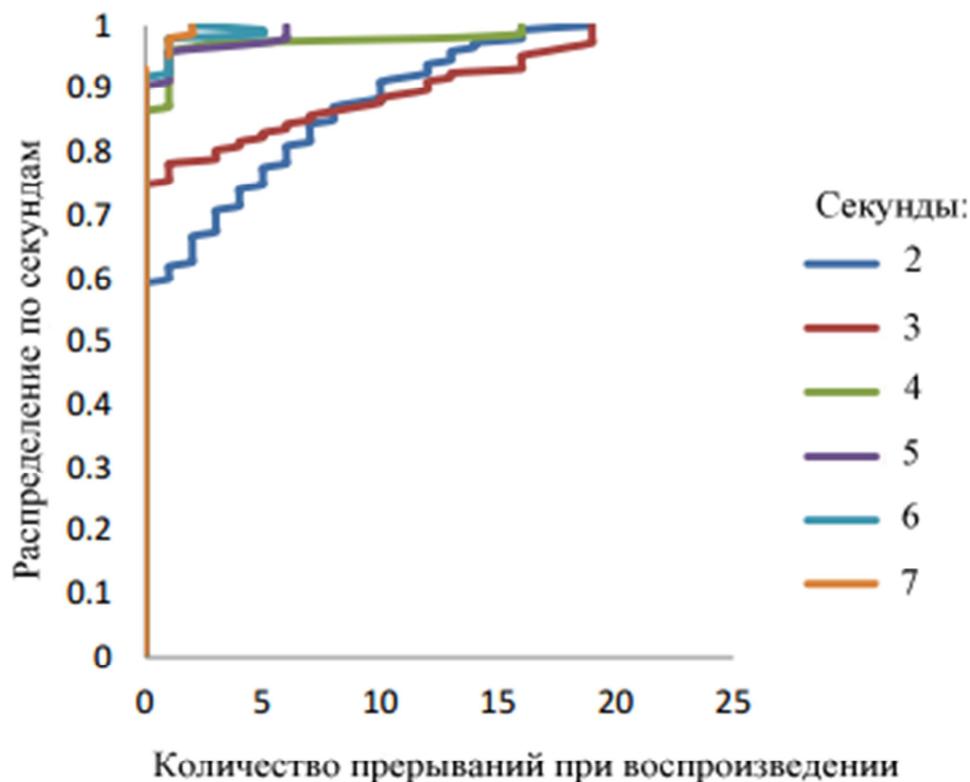


Рис. 4. Зависимость количества прерываний в воспроизведении видеопотока от времени, затраченного на составление формулы для определения идеального размера буфера данных

Таким образом, мы видим, что после 8-й секунды воспроизведения, вероятность прерывания видеопотока из-за малой пропускной способности канала резко сокращается. Эта вероятность находится в диапазоне 1–2 %.

Проиллюстрируем полученный способ интерполяции размеров входящего буфера на примере видео хостинга YouTube.

Для проведения тестов, аналогичных с изложенными выше, выбирается единственно видео с хостинга, и для него проводится сбор данных. Таким образом, после нескольких итераций мы имеем реальный размер буфера приложения на YouTube, идеальный размер буфера для определенного интервала изменений в ширине канала, а также предсказанный по формуле (1) размер буфера:

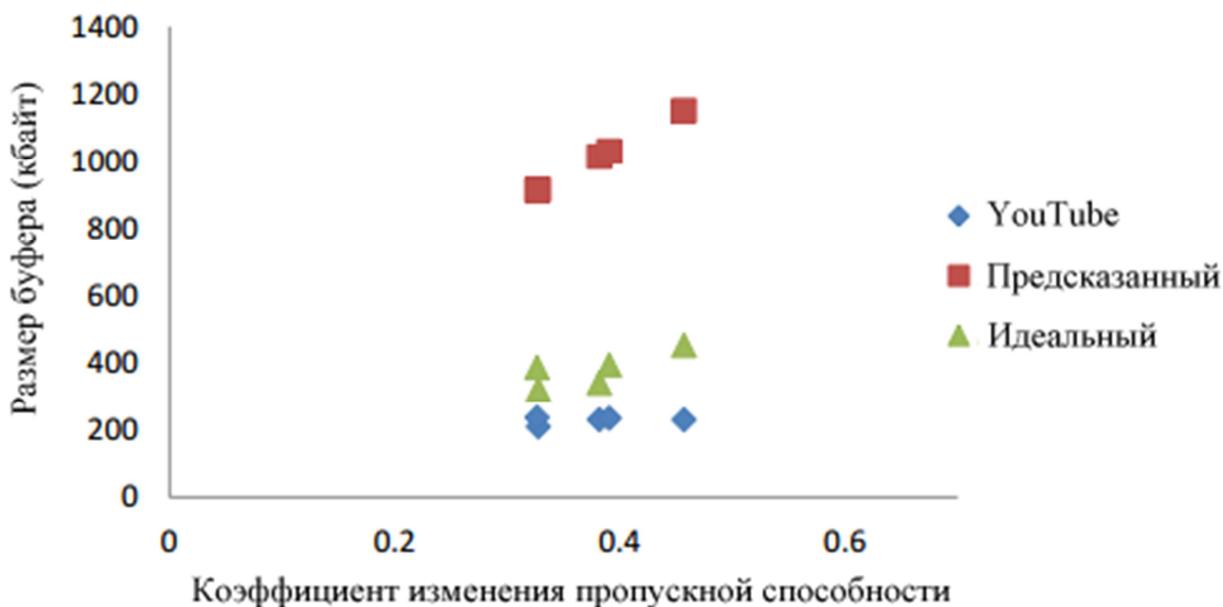


Рис. 5. Сравнение различных размеров буферов данных на примере веб-приложения YouTube

Как видно из графика, размер буфера, используемый YouTube, не зависит от скорости соединения пользователя, а значит, игнорирует комфортный userexperience. Также на графике представлен идеальный размер буфера, определенный на основе обработки статистических данных, и предсказанный размер буфера, определяемый по вычисленной нами ранее формуле. Последний обеспечивает нулевое количество прерываний, поэтому существенно превышает по размеру два других, но не превышает необходимых границ.

Очень важным является изучение вопроса, связанного с размером буфера входных данных, поскольку его значение, изменяемое вместе с изменением пропускной способности, в значительной степени сказывается на плавности воспроизведения и количестве нежелательных прерываний видео. Проведенная нами экспериментальная работа показывает, что размеры этого буфера должны изменяться при изменении пропускной способности.

Заключение

Мы построили специальную среду из программных продуктов Dummynet, Wireshark, а также серверного и клиентского приложений, написанных нами на языке высокого уровня Java, для эмуляции поведения пакетов и на основе нескольких итераций выяснили, что изменение в размере буфера должно происходить линейно и быть пропорциональным коэффициенту изменения пропускной способности.

Используя полученную нами формулу для идеального размера буфера, мы провели измерения на небольших интервалах воспроизведения и обнаружили, что точность нашего метода по динамическому определению размера буфера резко повышается спустя 8 секунд

воспроизведения. Таким образом, мы пришли к выводу, что достаточно точную формулу по определению идеального размера можно построить, наблюдая за изменениями в пропускной способности за первые 8 секунд воспроизведения. Этот подход рекомендуется использовать в мультимедийных приложениях, передающих видеоданные по сети, поскольку предложенный способ динамического изменения буфера данных повышает плавность воспроизведения.

В завершении мы провели ряд экспериментов по определению реакции буфера клиентского приложения YouTube на изменения в пропускной способности клиента и обнаружили, что размер буфера данных YouTube не реагирует на изменение пропускной способности канала передачи данных клиента.

Список литературы

1. Akshabi S., Begen A., Dovrolis C. An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP // Proceedings of the second annual ACM conference on Multimediasystems, ACM New York. – NY, USA, 2011.
2. Kim T., Ammar M. H. Receiver Buffer Requirement for Video Streaming Over TCP // Proceedings of Visual Communications and Image Processing, San Jose. – CA, USA, 2006.
3. Krasic C., Goel A., Walpole J. Low-Latency Adaptive Streaming over TCP // ACM Transactions on Multimedia Computing, Communications and Applications. – August 2008. – Vol. 4, № 3.
4. Nukhet O., Turhan T. On optimal receiver buffer size in adaptive Internet video streaming // Journal of Zhejiang University. – Turkey, 2006.
5. Zambelli A. IIS Smooth Streaming Technical Overview // Microsoft Corporation, Technical report. – MS, 2009.

Рецензенты:

Лукьянов Виктор Сергеевич, д-р техн. наук, профессор кафедры ЭВМ и систем Волгоградского государственного технического университета, г. Волгоград.

Муха Юрий Петрович, д-р техн. наук, профессор, заведующий кафедрой «Вычислительная техника» Волгоградского государственного технического университета, г. Волгоград.