

РАЗРАБОТКА СТЕНДА ДЛЯ ОЦЕНКИ ПРИМЕНИМОСТИ ТРАНСПОРТНЫХ ПРОТОКОЛОВ В ЗАДАЧАХ ОБРАБОТКИ ПОТОКОВОЙ ИНФОРМАЦИИ ДЛЯ СОЗДАНИЯ АДАПТИВНОЙ СИСТЕМЫ ПРЕОБРАЗОВАНИЯ ДАННЫХ

А. А. Большаков, И. В. Егоров, В. В. Лобанов, Д. В. Лачугин

*Кафедра «Автоматизация, управление, мехатроника»,
ФГБОУ ВПО «Саратовский государственный технический университет
им. Ю. А. Гагарина», г. Саратов; aabolshakov57@gmail.com*

Ключевые слова и фразы: программно-аппаратный стенд; протоколы транспортного уровня; распределенные информационные системы; сравнительный анализ.

Аннотация: Представлен сравнительный анализ протоколов транспортного уровня TCP, UDP и UDP-Lite. Приведены основные принципы работы данных протоколов и сформулированы их отличительные особенности. Создан программно-аппаратный стенд и предложена методика тестирования, позволяющая получить качественные и количественные результаты сравнения протоколов для работы с потоковой информацией. Получены результаты, позволяющие сформулировать адекватную оценку производительности и надежности работы распределенной системы при использовании определенного протокола, и сформированы рекомендации по их применению.

Введение

В настоящее время коммуникационные системы, представляющие сети передачи информации, находят все большее применение в различных областях деятельности. Распределенные информационные системы (ИС), позволяющие собирать, обрабатывать и хранить информацию [1 – 3], не являются исключением. Ядром таких систем являются коммуникационные протоколы, осуществляющие координацию процессов передачи массивов данных. Растущий объем информации, который необходимо обрабатывать с использованием распределенных систем с учетом требований функционирования в жестких временных ограничениях, обуславливает целесообразность работы с потоковой информацией. Поэтому актуальной является задача исследования коммуникационных протоколов сетевой подсистемы распределенной ИС.

Протоколы передачи информации различаются по степени общности решаемых задач на несколько уровней, которые образуют сетевую архитектуру. Системы в сети делятся на конечные узлы, между которыми осуществляется обмен массивами данных, и транзитные. Двухсторонний обмен информацией между парой смежных систем обеспечивается каналом связи. Каналы характеризуются скоростью информационного потока, задержкой передачи и вероятностью битовых ошибок. Если скорость поступления информации в узел превышает скорость ее отправки, то это приводит к переполнению буферов, что влечет потерю информации (перегрузку сети).

Протокол транспортного уровня занимает важное положение в сетевой архитектуре. Его основная задача – организация связи между узлами сети (host-to-host). Транспортный уровень должен компенсировать ненадежность передачи данных протоколами более низких уровней обработкой ошибок и заданием правил поведения для участников информационного обмена.

Традиционными протоколами транспортного уровня являются протоколы Transmission Control Protocol (**TCP**, протокол управления передачей) [4], User Datagram Protocol (**UDP**, протокол пользовательских дейтаграмм) [5], а также его модификация Lightweight User Datagram Protocol (**UDP-Lite**, облегченный протокол пользовательских дейтаграмм) [6]. При этом проводится работа по созданию современных протоколов, которые будут альтернативой традиционным UDP и TCP. Такими «молодыми» разработками являются Stream Control Transmission Protocol (**SCTP**, протокол передачи с управлением потоком) [7] и Datagram Congestion Control Protocol (**DCCP**, протокол управления перегрузкой для дейтаграмм) [8]. Оба протокола включают элементы TCP и UDP, и соответственно обладают частью достоинств и недостатков традиционных протоколов передачи. Обобщенные свойства данных транспортных протоколов сведены в табл. 1.

Особенностью *протокола UDP* является необходимость при разработке приложений реализовывать механизмы повторной передачи данных, обработки неправильной последовательности и потери пакетов. Рассматриваемый протокол позволяет использовать относительно простые заголовки, не требующие значительных вычислительных и ресурсных затрат от устройств сетевой передачи. Не тратится время на установку соединения («рукопожатия»), подтверждение доставки, контроль перегрузки сети. В результате, обмен пакетами происходит очень быстро, однако, отсутствие контроля перегрузок, возрастающего объема неконтролируемой нагрузки при использовании сетевой инфраструктуры совместно с другими сервисами создает потенциальную опасность перегрузки сети, которая ведет к значительному падению ее производительности. Многолетняя практика показывает, что протокол UDP хорошо зарекомендован для ширококов-

Таблица 1

Характеристики транспортных протоколов

Наименование характеристики	Протокол				
	TCP	UDP	UDP-Lite	SCTP	DCCP
Установка соединения между отправителем и получателем	Да	Нет	Нет	Да	Да
Неупорядоченное следование пакетов	Нет	Да	Да		Нет
Упорядоченное следование пакетов	Да	Нет	Нет		Да
Контроль перегрузки сети как часть протокола транспортного уровня					Да
Подтверждение доставки пакетов					Нет
Возможность управления накоплением	Нет	Да	Да	Да	
Контроль дублирования пакетов				Нет	
Сохранение границ сообщения	Нет	Да	Да	Нет	Нет
Контроль ошибок в байтах (Cyclic Redundancy Check – CRC)	Да				
Контроль частичной целостности пакета	Нет	Нет	Нет	Да	Нет
Многопоточная передача					
Поддержка многоинтерфейсных узлов (multihoming)			Нет	Нет	

щательного трафика, сервисов потоковой передачи аудио- и видеоинформации, в протоколах сигнализации в **VoIP** (Voice over IP) сетях.

Протокол TCP реализует обработку тайм-аутов, отправку и прием подтверждений, повторную передачу данных при их потере, определение и отброс дублирующих пакетов. Вышеперечисленные особенности протокола TCP кроме положительных последствий имеют и отрицательные. Так, тройное «рукопожатие», в свою очередь, является уязвимым местом протокола для проведения атак в целях переполнения запросами на подключение (SYN-атаки, Synchronize) для «отказа от обслуживания» (Denial of Service – DoS). Если отправитель не получает подтверждение об успешной доставке, то следующие данные не пересылаются, а повторяется попытка доставить потерянный блок данных, что может потенциально создать задержку до нескольких минут. Протокол ориентирован на работу с потоком байт, поэтому от разработчика требуется реализация процедуры формирования кадров сообщения. В противном случае узел-приемник может в момент чтения получить блок данных большего или меньшего размеров, чем переданный. Использование протокола связано со значительными вычислительными и ресурсными затратами, так как необходимо сохранять порядок следования байт, поддерживать функционирование механизма «скользящего окна» и т.п. Поэтому передача пакетов по протоколу TCP ведет к ощутимым временным задержкам.

Традиционно в качестве транспортного уровня для передачи потоковой видео- и аудиоинформации используется протокол UDP, который содержит поле контрольной суммы в заголовке, покрывающее всю полезную нагрузку. На приемной стороне проверка поврежденных байтов после декодирования канала происходит на физическом уровне. Это приводит к тому, что пакеты с ошибкой в байтах отбрасываются. Однако некоторые кодеки, например Adaptive Multi-Rate, Internet Low Bit Rate Codec для VoIP или Universal Mobile Telecommunications System (UMTS) сетей, видеокодеки H.263+, H.264, MPEG-4 достаточно устойчивы к небольшим ошибкам в относительно малой части фрейма. Они разработаны так, что ошибки в данных для них будут предпочтительней отбрасывания пакетов целиком. В Internet Engineering Task Force (IETF) разработан новый транспортный *протокол UDP-Lite*, в котором обеспечивается поддержка контрольных сумм с опциональным частичным покрытием данных. При использовании данной опции пакет делится на две части: чувствительную к ошибкам (обнаруживаются контрольной суммой) и нечувствительную к ним (не обнаруживаются суммой). Ошибки в нечувствительной к ним части не будут приводить к отбрасыванию пакетов транспортным уровнем принимающего конечного хоста. По сравнению с протоколом UDP, неполные контрольные суммы UDP-Lite обеспечивают дополнительную гибкость приложений, для которых требуется определить часть данных в пакетах, как нечувствительную к ошибкам, что достаточно актуально при потоковой передаче данных.

Причем, UDP-Lite – модифицированный UDP-протокол, различие в заголовках проиллюстрировано на рис. 1. Поле длины из пакета UDP заменено на поле размера покрытия данных контрольной суммой. Однако имеются некоторые ограничения: необходимо, чтобы заголовок был защищен; защищаемые данные должны располагаться в начале пакета. Данная замена допустима, так как информация о размере пакета UDP может быть обеспечена модулем IP так же, как это делается для TCP.

Протокол DCCP использует двунаправленные соединения с управлением перегрузкой. Основное применение он может найти в приложениях, которые реализуют поточную схему TCP, однако, имеют приоритет для своевременной доставки данных. Протокол DCCP разработан как альтернатива UDP для приложений, которым необходима негарантированная, но своевременная доставка данных, высокая скорость работы и реализованные на транспортном уровне механизмы для отслеживания перегрузок в сети без необходимости их создания

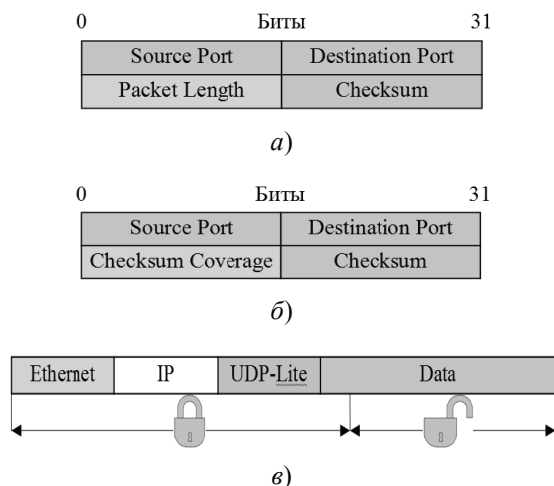


Рис. 1. Заголовок пакета UDP (а), UDP-Lite (б), схема Ethernet-фрейма с частичным покрытием данных CRC (в)

на уровне приложений. Стандарт протокола развивается, ведутся работы в области поддержки многопоточной передачи дейтаграмм, а также по созданию механизмов отслеживания перегрузок в сети. В текущей редакции поддерживается несколько вариантов: TCP-подобное сокращение вдвое окна передачи и алгоритм управления перегрузкой, базирующийся на уравнении для минимизации резких изменений скорости передачи. Поддержка протокола в Linux является экспериментальной, например, присутствуют проблемы с фрагментацией данных объема больше, чем размер Maximum Transmission Unit (MTU). Поэтому в настоящей статье протокол DCCP далее не рассматривается.

Протокол SCTP разработан и стандартизирован Internet Engineering Task Force (IETF) в 2000 году, однако, до сих пор недостаточно активно применяется. Он создает двухстороннюю ассоциацию между двумя конечными точками и обеспечивает возможность работы с несколькими потоками каждой паре конечных узлов, а также поддержку концепции многоинтерфейсного узла на транспортном уровне (рис. 2). При этом данный протокол можно считать альтернативой TCP, так как гарантированы доставки и сохранение порядка следования пакетов, также могут быть настроены частичное упорядочивание, или неупорядоченная доставка. Такой подход позволяет устранить задержку. Имеется механизм контроля перегрузки и восстановления потерянных пакетов. Концепция SCTP обуславливает работу не с потоком неструктурированных байт, а пакетов, сохраняя в них границы сообщений. Однако в текущей редакции протокола, несмотря

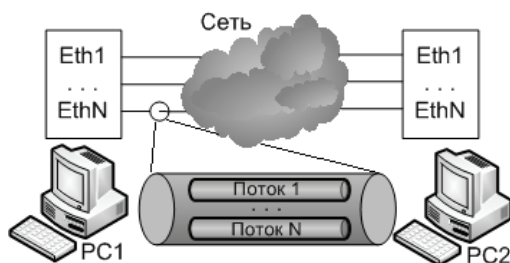


Рис. 2. Мультиточечная SCTP ассоциация конечных узлов со множественными интерфейсами (multihoming)

на то что заявлена поддержка многоинтерфейсных узлов, в действительности, рассматриваемый подход не поддерживает распределение нагрузки по нескольким интерфейсам. Поэтому технология только обеспечивает избыточность и создает альтернативные направления. Поддержка протокола в Linux (ядро 2.6.11 и выше) имеется, однако выключена по умолчанию. Для Windows (Vista и выше) включена в виде дополнительного коммерческого пакета независимого поставщика либо в составе некоторых программных пакетов.

В нашей стране и за рубежом не прекращаются работы по модификации традиционных транспортных протоколов, например TCP с линейным сетевым кодированием (TCP/NC) [9], и по созданию на их базе новых улучшенных протоколов ARTCP [10].

Постановка задачи

Рассмотрено повышение эффективности работы распределенных информационных систем сбора и обработки информации адаптацией к изменяемым условиям среды передачи, а также управляемого сжатия массивов передаваемых данных в виде информационных потоков. Для ее достижения требуется решить следующие задачи:

а) анализ функционирования коммуникационной подсистемы распределенной ИС при использовании различных транспортных протоколов;

б) выявление статистических характеристик потоковой информации и определение на их основе класса вейвлетов, применимых для обработки потоков;

в) разработка концепции и алгоритмов управляемого сжатия потоковой информации, а также адаптации коммуникационной подсистемы к изменяющимся условиям передачи.

Выявление характеристик потоковой информации, определение класса вейвлетов, а также разработка алгоритмов управляемого сжатия адаптивной ИС не входят в рамки настоящей статьи, они являются предметом наших последующих исследований.

Целью настоящей работы является решение первой из вышеперечисленных задач, что связано с созданием стенда для исследования ряда стандартизированных протоколов транспортного уровня модели OSI для оценки степени применимости их в распределенных системах сбора и обработки потоковой информации. В частности, необходимо исследовать такие протоколы, как TCP, UDP, UDP-Lite. Требуется экспериментально определить особенности данных протоколов, являющиеся ограничивающими факторами при проектировании распределенных систем обработки потоковых данных. Исследованы зависимости от применения определенного протокола в различных условиях среды передачи. Проанализирована «логика» функционирования протоколов и ситуации, при которых предпочтительно использовать определенный протокол.

Настоящее исследование определяет качественные и количественные результаты сравнения традиционных протоколов транспортного уровня для работы с потоковой информацией, что позволяет находить степень применимости определенного протокола в распределенной системе сбора и обработки информации [3]. Полученные результаты дают возможность сформулировать адекватную оценку производительности и надежности работы предлагаемой системы.

Транспортные протоколы SCTP и DCCP достаточно «молодые», над их усовершенствованием и стандартизацией проводится дальнейшая работа. При этом имеется потенциал для их применения не только в системах, работающих с потоковыми данными.

Описание программно-аппаратного стенда для тестирования

Для проведения тестирования использована схема стенда, приведенная на рис. 3. Она состоит из двух компьютеров PC1 и PC2 с операционной системой Linux с ядром версии 3.7.10. На PC1 установлена передающая часть тестового приложения, PC2 – принимающая. Рабочие машины соединяют три узла: два маршрутизатора R1 и R2, стоящие непосредственно после компьютеров, и коммутатор SW1. Тестовый стенд работает с шириной канала равной 100 Мбит для всех участков сети. Для эмуляции нагрузки в сети используется генератор сетевого трафика Spirent SmartBits SMB-600B [11]. Использовались Ethernet-кадры с разными размерами: 128, 256, 512, 1024, 1280, 1518. В частности, следует проводить тесты с кадрами минимального и максимального допустимого для среды размера, чтобы получить полный набор характеристик работы. Анализируется работа только в IPv4 сети.

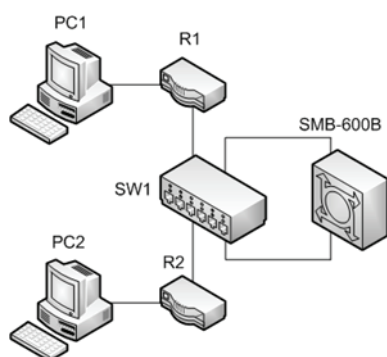


Рис. 3. Стенд для тестирования

Предложен набор тестов, которые могут быть применены для измерения параметров производительности. Для анализа полученных пакетов проверяются:

- размер принятых кадров на предмет соответствия ожидаемому значению;
- порядковые номера, включаемые в передаваемые кадры;
- штампы времени отправки и приема кадров.

Порядковые номера проверяются в принимаемых кадрах для определения чисел: отброшенных фреймов; доставленных с нарушением порядка следования пакетов; полученных дубликатов; пропусков в порядковых номерах принятых пакетов.

Реализация тестов приведена в упрощенной форме: использовался один логический поток данных с одним фиксированным IP-адресом отправителя и одним фиксированным IP-адресом получателя. Следует отметить, что реальные сети не ограничиваются единственным потоком данных. Для создания повышенной нагрузки на сеть использован генератор сетевого трафика Spirent SmartBits SMB-600B, который позволяет формировать произвольное число потоков дополнительного интернет-трафика различного типа и интенсивности, моделируя при этом работу типовых служб и сервисов компьютеров, подключенных к реальной сети. Пропускная способность сети передачи данных непосредственно зависит от уровня загрузки сети, поэтому для тестов с использованием генератора трафика можно управлять процентом загруженности передающей магистрали.

Длительность проведения тестов 120 с. В таблице 2 приведен пример тестового UDP-кадра с произвольными данными в качестве полезной нагрузки.

Проведены тесты согласно следующей методике.

1. Измерение величины задержки. Для этого формируется и передается поток кадров заданного размера с определенной полосой, адресованных одному получателю. В поток включается особый кадр – идентификационная метка. Фиксируется время завершения передачи метки T_A на передающей стороне. Логика приемной части распознает метку в потоке кадров и фиксирует время завершения приема кадра с меткой T_B . Задержка T вычисляется по формуле

$$T = T_B - T_A.$$

Тестовый UDP кадр в среде Ethernet

Поле	Смещение и данные (hex)		Описание
Ethernet-заголовок	00	00 C2 66 C9 57 E3	MAC-адрес получателя
	06	02 AC 01 67 8D 3B	MAC-адрес отправителя
	12	08 00	Тип инкапсулируемого протокола (в данном случае IPv4)
IP-заголовок	14	45	Версия IP – 4; размер заголовка – 5 четырехбайтных слова
	15	00	Поле TOS (тип обслуживания)
	16	00 2E	Общий размер
	18	00 00	ID-фрагмент пакета
	20	00 00	Флаги контроля фрагментации и смещение фрагмента
	22	0A	Время жизни пакета (TTL)
	23	11	Тип инкапсулируемого протокола (UDP)
	24	C4 8D	Контрольная сумма IP-заголовка
	26	AC 11 01 64	IP-адрес отправителя
	30	AC 11 01 C8	IP-адрес получателя
UDP-заголовок	34	C0 20	Порт отправителя
	36	0D A7	Порт получателя
	38	00 1A	Размер сообщения UDP
	40	00 00	Контрольная сумма
UDP-данные	42	00 01 02 03 04 05	Произвольные данные
	48	06 07 08 09 0A 0B	
	54	0C 0D 0E 0F	

2. Измерение частоты потери кадров. Формируется и передается поток с заданными числом кадров C_{tr} и размером кадра. После передачи определяется число принятых кадров C_r и вычисляется величина потерь

$$C = ((C_{tr} - C_r)100) / C_{tr}.$$

3. Измерение влияния потери кадров на скорость передачи информации.

4. Определение вероятности возникновения поврежденного кадра, который отбрасывается для обеспечения безошибочной передачи данных. Упрощенно процесс потоковой передачи представлен на рис. 4. Для простоты предполагаем, что первый пакет генерируется во время 0. Более поздние пакеты создаются с постоянной скоростью, равной генерации потока данных первичными преобразователями информации.

На рис. 4 иллюстрируется число пакетов $G(t)$, генерируемых за некоторое время, то есть $G(t) = \mu t$. Кривой $A(t)$ обозначим число пакетов, достигших приемную сторону за некоторое время t . Данный показатель ограничен скоростью генерации пакетов $A(t) \leq G(t)$. Характеристикой $B(t)$ обозначим число пакетов,

которые могут быть обработаны на приемной стороне. Однако в распределенной системе передачи данных обязательно присутствует некоторая задержка τ , поэтому $B(t) = \mu(t - \tau)$. В общем случае, в систему должны приходить актуальные пакеты $N(t) = A(t) - B(t)$. Отрицательная величина данного показателя показывает, что пакеты «запоздали» (так как если используется транспортный протокол TCP, то гарантируется надежность доставки пакетов), либо потерялись, что является нормальным поведением для UDP. Число запоздавших пакетов обозначается n_L .

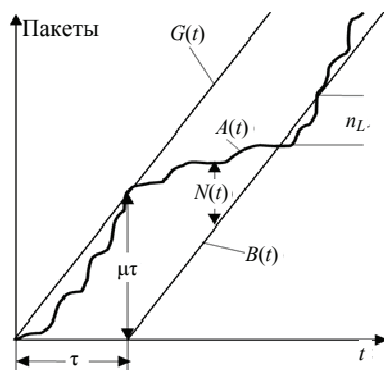


Рис. 4. Процесс потоковой передачи

Описание и анализ результатов тестирования

На рисунке 5 приведен график зависимости времени задержки в доставке пакетов от размера Ethernet-кадров. Результат приведен при нагрузке на сеть, равной 80 %. Из рисунка видно, что при малых размерах Ethernet-кадров задержки распространения пакетов UDP, TCP и UDP-Lite не отличается. При увеличении размера фрейма задержки становятся ощутимы: пакеты TCP обрабатываются медленнее, а различие в скорости обработки UDP и UDP-Lite практически отсутствует. Время доставки пакетов определяется формулой

$$t = t_{cr} + S/B + l/c + t_c,$$

где t_{cr} – время формирования и выдачи пакета из памяти в сетевой интерфейс; S – размер пакета; B – пропускная способность интерфейса; l – длина линии связи; c – время распространения сигнала по линии связи; t_c – время обработки пакета приемной машиной. Из этого следует, что временные задержки имеют выраженную линейную зависимость от размера пакета, а отклонение во времени передачи TCP и UDP вызвано тем, что обработка TCP пакета и заголовка, в частности, требует больше затрат вычислительных ресурсов и ресурсов памяти как у отправляющей, так и у принимающей машины.

На рисунке 6 показан график зависимости вероятности возникновения поврежденных пакетов от их размера, когда из-за поврежденных битов пакет отбра-

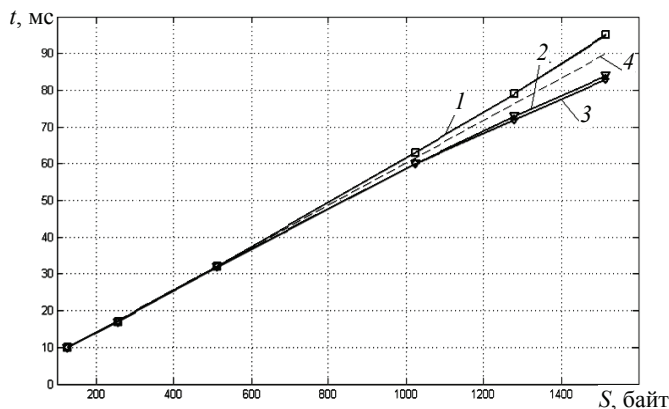


Рис. 5. График зависимости времени задержки от размера Ethernet-кадра:
1 – TCP; 2 – UDP; 3 – UDP-Lite; 4 – Linear

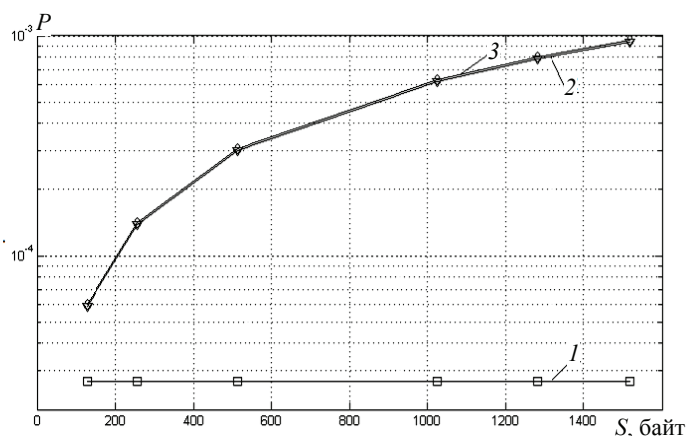


Рис. 6. Зависимость вероятности возникновения поврежденных пакетов от их размера:
 1 – TCP; 2 – UDP; 3 – UDP-Lite

сывается. Очевидно, что вероятность для TCP и UDP-пакетов становится больше с увеличением размера Ethernet-кадра, что логично, так как увеличивается число битов, которые могут быть повреждены. При этом протокол UDP-Lite «терпим» к увеличению размера фрейма, особенно, если контрольная сумма CRC-16 покрывает только заголовок. Отмечено, что в протоколах UDP и TCP, если в заголовке H , либо в полезной нагрузке D появятся ошибки в битах, то пакет считается поврежденным и будет отброшен. Вероятность этого определяется выражением

$$P_{\text{UDP, TCP}} = 1 - (1 - P_{\text{BER}})^{H+D}.$$

где H, D – число битов в заголовке и поле данных соответственно; P_{BER} – вероятность появления ошибки в потоке битов; BER (bit error rate) – частота ошибок в битах потока данных, идущего поверх коммуникационного канала, характеристика качества передаваемых данных, в данном случае, по интерфейсу Ethernet. Стандарт IEEE 802.3 [12] определяет величину P_{BER} равной $1E-10$.

В протоколе UDP-Lite пакет будет отброшен, только если поврежден заголовок, тогда вероятность потери пакета в данном случае определяется выражением

$$P_{\text{UDP-Lite}} = 1 - (1 - P_{\text{BER}})^H.$$

Таким образом, вероятность $P_{\text{UDP-Lite}}$ существенно меньше $P_{\text{UDP, TCP}}$, поэтому протокол UDP-Lite показывает большую эффективность, так как при увеличении размера кадра уровень потерь не увеличивается.

Значительное увеличение нагрузки на сеть приводит к возможным потерям пакетов, например, из-за переполнения очередей или буферов в магистральных устройствах. Увеличение потерь пакетов негативно влияет на производительность сети (рис. 7), особенно для TCP-протокола. Кривая достаточно резко падает, что связано с особенностью работы TCP: потери пакетов оцениваются как признак перегруженности сети. Для устранения перегруженности протокол уменьшает размер передаваемого буфера вдвое, увеличивает время ожидания и отправляет дубликат пакета. Алгоритм приводит к тому, что скорость передачи искусственно снижается в десятки раз. Для UDP и UDP-Lite протоколов потери пакетов в сети не так существенны в аспекте скорости передачи. Снижение производительности наблюдается, однако незначительное.

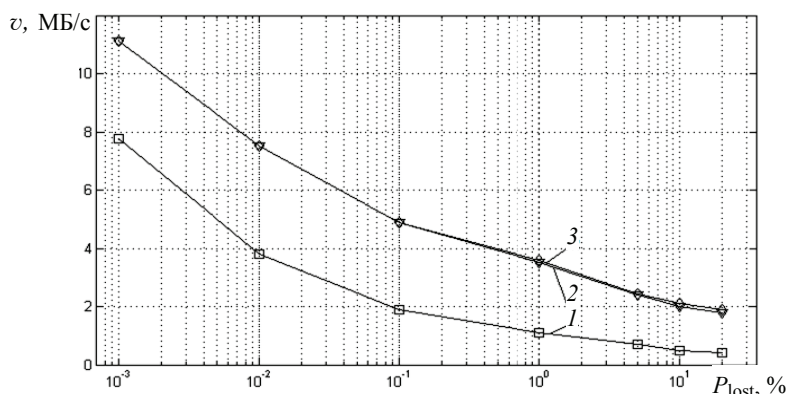


Рис. 7. Зависимость скорости передачи данных от вероятности потери пакета:
1 – TCP; 2 – UDP; 3 – UD-Lite

Заключение

Приведены краткая характеристика традиционных транспортных протоколов, а также результаты экспериментов по их применению для передачи потоковых данных. Выбрать и использовать только один из протоколов на основе результатов эксперимента не представляется возможным, так как каждый из протоколов позволяет в полной мере использовать преимущества лишь в определенных условиях работы. Поэтому выбор конкретного протокола должен осуществляться динамически, в зависимости от сложившихся условий работы.

На основе полученных результатов можно сформулировать следующие рекомендации. Первоначально предлагается использовать протокол TCP. Если распределенная система сбора и обработки данных определит, что быстродействия протокола «не хватает для комфортной работы» системы в заданном темпе времени, то следует переключиться на транспортный протокол UDP. Такой выбор увеличит быстродействие из-за возможных потерь данных. Пакеты, которые получит система, будут актуальными и без ошибочных данных. Если система определит, что быстродействия протокола UDP не хватает, то следует отдать предпочтение UDP-Lite. Это позволит повысить быстродействие, причем система сбора и обработки данных должна работать не только с пропусками информации, но и уметь определять поврежденные участки в пакетах и корректно их обрабатывать.

Список литературы

1. Благовещенская, М. М. Система автоматического регулирования с цифровой видеокамерой / М. М. Благовещенская, Я. В. Иванов // Вестн. Тамб. гос. техн. ун-та. – 2010. – Т. 16, № 4. – С. 776 – 779.
2. Егоров, А. Ф. Нечеткая система управления показателями качества продукции первичной переработки нефти / А. Ф. Егоров, П. Г. Михайлова, Д. М. Хунг // Вестн. Тамб. гос. техн. ун-та. – 2013. – Т. 19, № 4. – С. 757 – 763.
3. Transmission Control Protocol [Электронный ресурс]. – Режим доступа : <http://tools.ietf.org/html/rfc793> (дата обращения: 15.05.2014).
4. User Datagram Protocol [Электронный ресурс]. – Режим доступа : <http://tools.ietf.org/html/rfc768> (дата обращения: 15.05.2014).
5. The Lightweight User Datagram Protocol (UDP-Lite) [Электронный ресурс]. – Режим доступа : <http://tools.ietf.org/html/rfc3828> (дата обращения: 15.05.2014).
6. Stream Control Transmission Protocol [Электронный ресурс]. – Режим доступа : <http://tools.ietf.org/html/rfc4960> (дата обращения: 15.05.2014).

7. Datagram Congestion Control Protocol (DCCP) [Электронный ресурс]. – Режим доступа : <http://tools.ietf.org/html/rfc4340> (дата обращения: 15.05.2014).
 8. Network Coding Meets TCP: Theory and Implementation / J. K. Sundararajan [et al.] // Proceedings of the IEEE. – 2011. – Vol. 39. – No. 3. – P. 490 – 512.
 9. Алексеев, И. В. Адаптивная схема управления потоком для транспортного протокола в сетях с коммутацией пакетов : дис. ... канд. физ.-мат. наук : 05.13.17 / Алексеев Игорь Вадимович. – Ярославль, 2000. – 141 с.
 10. Техническая диагностика по сигнальной информации на примере мониторинга состояния глубинного насоса / А. А. Большаков [и др.] // ММТТ-26 : сб. тр. XXVI Междунар. науч. конф., Ангарск – Иркутск, 9 – 13 сент. 2013 г. : в 2 ч. / под ред. А. А. Большакова. – Ангарск ; Иркутск, 2013. – Ч. 2. – С. 40 – 42.
 11. Portable High-Density Network Performance Analysis System Smart Bits 600B [Электронный ресурс]. – Режим доступа : http://www.spirent.com/~media/Datasheets/Broadband/Obsolete_SMB-TM/SmartBits%20600B.pdf (дата обращения: 15.05.2014).
 12. IEEE Standard 802.3 [Электронный ресурс]. – Режим доступа : http://read.pudn.com/downloads139/sourcecode/others/600448/ieee8023_2005/802.3-2005_section1.pdf (дата обращения: 15.05.2014).
-

Designing a Stand for a Comparative Analysis of Transport Layer Protocols for Information Flow Processing to Create an Adaptive Data Conversion System

A. A. Bolshakov, I. V. Egorov, V. V. Lobanov, D. V. Lachugin

*Department «Automation, Control, Mechatronics»,
Yuri Gagarin State Technical University of Saratov;
aabolshakov57@gmail.com*

Key words and phrases: comparative analysis; distributed information systems; software and hardware stand; transport layer protocols; TCP; UDP; UDP-Lite.

Abstract: A comparative analysis of transport layer protocols such as TCP, UDP and UDP-Lite is proposed in this paper. Some basic principles of these protocols and their features are described in the article. Stand for testing was constructed and described, and technique of testing is proposed. This technique allows obtaining qualitative and quantitative data from the comparison of these protocols to use them with streaming data. These results allow assessing performance and reliability of a distributed system of collection and processing data using a particular protocol. At the end of the article we give some recommendations for transport protocols application.

References

1. Blagoveshchenskaya M.M., Ivanov Ya.V. *Transactions of the Tambov State Technical University*, 2010, vol. 16, no. 4, pp. 776-779.
2. Egorov A.F., Mikhailova P.G., Khung D.M. *Transactions of the Tambov State Technical University*, 2013, vol. 19, no. 4, pp. 757-763.
3. <http://tools.ietf.org/html/rfc793> (accessed 15 May 2014).
4. <http://tools.ietf.org/html/rfc768> (accessed 15 May 2014).
5. <http://tools.ietf.org/html/rfc3828> (accessed 15 May 2014).
6. <http://tools.ietf.org/html/rfc4960> (accessed 15 May 2014).
7. <http://tools.ietf.org/html/rfc4340> (accessed 15 May 2014).

8. Sundararajan J.K., Shah D., Me'dard M., Jakubczak S., Mitzenmacher M., Barros J. *Proceedings of the IEEE*, 2011, vol. 39, no. 3, pp. 490-512, doi: 10.1109/JPROC.2010.2093850.

9. Alekseev I.V. *PhD dissertation (Physics and Mathematics)*, Yaroslavl', 2000, 141 p.

10. Bol'shakov A.A., Glazkov V.P., Lachugin D. V., Lobanov V.V., in Bol'shakov A.A. (Ed.) *MMTT-26*, Proceedings of the XXVI International Conference, vol. 2 of 2, Angarsk, Irkutsk, 2013, pp. 40-42.

11. http://www.spirent.com/~media/Datasheets/Broadband/Obsolete_SMB-TM/SmartBits%20600B.pdf (accessed 15 May 2014).

12. http://read.pudn.com/downloads139/sourcecode/others/600448/ieee8023_2005/802.3-2005_section1.pdf (accessed 15 May 2014).

Entwicklung des Standes für die Einschätzung der Anwendbarkeit der Transportprotokolle in den Aufgaben der Bearbeitung der Strominformation für die Bildung des anpassungsfähigen Systems der Umgestaltung der Daten

Zusammenfassung: Es ist die vergleichende Analyse der Protokolle des Transportniveaus TCP, UDP und UDP-Lite dargelegt. Es sind die Hauptprinzipien der Arbeit dieser Protokolle angeführt und es sind ihre eigentümlichen Besonderheiten abgefasst. Es ist der Hardware-Softwarestand geschaffen und es ist die Methodik der Prüfung angeboten, die die qualitativen und quantitativen Ergebnisse des Vergleiches dieser Protokolle für die Arbeit mit der Strominformation zu bekommen erlaubt. Es sind die Ergebnisse, die die adäquate Einschätzung der Produktivität und der Zuverlässigkeit der Arbeit des verteilten Systems unter Anwendung von einem bestimmten Protokoll abzufassen erlauben, und es sind die Empfehlungen nach ihrer Anwendung gebildet.

Elaboration du stand pour une analyse comparative des protocoles de transport dans le processus du traitement de l'information de flux pour la création du système adaptatif de la conversion des données

Résumé: Est présentée une analyse comparative des protocoles du niveau TCP, UDP et UDP-Lite. Sont cités les essentiels principes du fonctionnement de ces protocoles, sont formulées leurs particularités. Est posée la méthode des tests permettant de recevoir les résultats qualitatifs et quantitatifs de la comparaison des protocoles pour le travail avec l'information de flux. Sont reçus les résultats permettant de formuler une estimation adéquate de la productivité et de la sécurité du fonctionnement du système; sont formulées les recommandations de l'application.

Авторы: *Большаков Александр Афанасьевич* – доктор технических наук, профессор кафедры «Автоматизация, управление, мехатроника»; *Егоров Игорь Владимирович* – кандидат технических наук, доцент кафедры «Автоматизация, управление, мехатроника»; *Лобанов Владимир Васильевич* – кандидат технических наук, доцент кафедры «Автоматизация, управление, мехатроника»; *Лачугин Дмитрий Вячеславович* – аспирант, ассистент кафедры «Автоматизация, управление, мехатроника», ФГБОУ ВПО «Саратовский государственный технический университет им. Ю. А. Гагарина», г. Саратов.

Рецензент: *Иващенко Владимир Андреевич* – доктор технических наук, старший научный сотрудник, ученый секретарь Института проблем точной механики и управления РАН, г. Саратов.