

Систематическая отладка передачи данных в стандарте HSDPA

69

ИЗМЕРЕНИЯ

измерения

Ричард Маруайр

Мобильные устройства следующего поколения, поддерживающие HSDPA, обещают обеспечить абонентам сетей UMTS (стандарт сетей 3G) скорость загрузки данных в 3,6 Мбит/с. Однако, в отличие от ныне работающих услуг передачи данных, эта функция доступна только при использовании мобильного устройства в связке с ПК. Это может привести к тому, что потребуются глубокое понимание процесса и владение новыми методиками для обеспечения потребностей клиента.

Мобильная система передачи данных

На рис. 1 показана система, предлагаемая для тестирования скорости передачи данных на стадии разработки. Она достаточно проста, и эмулирует условия, близкие к реальным, при этом давая определенный полезный при настройке максимальной скорости передачи данных контроль над переменными в системе. Вместо такой части реальной системы как Интернет используется выделенный сервер, отсекающий неконтролируемые задержки, существующие в реальной сети. Несмотря на усеченную версию сети, можно заметить серьезное падение скорости передачи данных, что заставляет нас обсудить компоненты нашей системы.

Протокольный уровень в системе

На рис. 2 показана упрощенная диаграмма уровней протоколов, задействованных в нашей системе. Мы можем наблюдать снижение скорости

передачи данных, которое происходит на многих уровнях. Те из них, которые могут являться источником проблем, выделены желтым цветом и описаны в данной статье. Мы постараемся увеличить скорость передачи данных, скорректировав минимальное количество уровней.

Режим тестирования РЧ канала передачи данных

Мы начнем с наименее сложного уровня, что позволит снизить количество проблемных взаимодействий и оценить работу ядра системы без добавления на этом этапе новых уровней. Для устройств HSDPA применим режим тестирования РЧ канала передачи данных (**Radio Bearer Test Mode**), чтобы оценить уровни, общие только для телефона и сотовой сети. В процессе передачи данных тест оценивает работу только РЧ-части и контроллера доступа к данным (MAC), отвечающих за установление соединения, необходимого нам для оценки системы на самом низком уровне. В данной статье предполагается, что мобильное устройство имеет цельный дизайн РЧ-части, поэтому при использовании тестового режима для тестирования MAC необходимо повысить мощность каналов данных нисходящей линии связи, чтобы устранить возможность ограничения скорости загрузки данных, вызванную РЧ-частью. На рис. 3 показано взаимодействие протоколов в режиме тестирования с упором на проверку работы контроллера доступа к данным (MAC) перед переходом к другим уровням системы. Кроме того, необходимо быть уверенным, что эмулятор сети обладает достаточной гибкостью и полосой пропускания для максимально возможной нагрузки MAC.



Рис. 1. Предлагаемая система для тестирования скорости передачи данных на стадии разработки

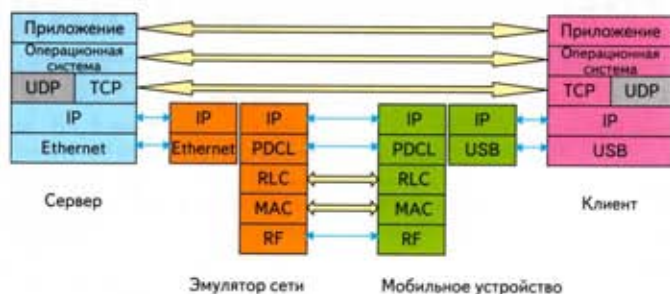


Рис. 2. Упрощенная диаграмма уровней протоколов

Итак, если нам удалось передать данные со скоростью 3,6 Мбит/с в режиме тестирования РЧ-канала, касающегося взаимодействия телефона с базовой станцией, мы можем быть уверены в этом уровне и сможем добавлять новые компоненты. Требуется сохранить все те настройки эмуляции сети, с которыми осуществлялось тестирование РЧ-канала, а затем присоединить к системе «компьютер-клиент и компьютер-сервер». Наличие в системе архитектуры «клиент — сервер» позволяет нам протестировать следующие уровни нашей системы, используя для этого протокол UDP (User datagram protocol).

UDP-флуд

На рис. 4 показан следующий шаг в решении наших задач, на котором используется так называемый флуд по протоколу UDP. Создание UDP-флуда — это процесс, инициированный сервером, суть которого состоит в том, что сервер отправляет данные напрямую на IP-адрес клиента, но поскольку эти данные не касаются какой-либо конкретной услуги, на клиентской машине нет никаких приложений, «слушающих» порт UDP, и вся информация просто отправляется в корзину. Флуд по UDP используется для тестирования RLC на скорости в 3,6 Мбит/с перед тестированием первого по-настоящему пользовательского приложения — потокового видео.

Потоковое видео

Серверы потокового видео используют UDP при отправке видеоданных, что позволяет нам на следующем этапе отладки оценить взаимодействие клиента с сервером, а также подключения внутри системы. Так как серверы соединяются по протоколу IP, следует оценить соединение клиента с сервером с помощью прямого подключения по кабелю LAN-X или через концентратор, пе-



Рис. 4. UDP-флуд

ред тем как включить всю систему, как показано на рис. 5. Опыт показывает, что запуск процесса передачи потокового видео не так уж прост, поэтому лучше сначала создать прямое подключение «клиент — сервер», настроить все параметры, убедиться в корректности работы и только потом запустить этот сервис по протоколу HSDPA.

Запуск этого приложения с использованием мобильной среды передачи данных, скорее всего, приведет к падению скорости ниже 3,6 Мбит/с, что сигнализирует о необходимости дополнительной настройки приложений. Самым быстрым способом проверить, лежит ли наша проблема в плоскости приложений — это создать журнал (лог), записывающий скорость передачи данных при передаче более чем одного фильма одновременно. Используя в качестве средства измерения соответствующую программу в рамках эмулятора сети или ту, что есть в операционной системе, можно оценить, добились ли мы своей цели — скорости 3,6 Мбит/с. Если скорость передачи одного фильма не достигает 3,6 Мбит/с, но при передаче нескольких фильмов одновременно скорость именно такова, значит, нам удалось определить, что на этом уровне система вполне способна передавать данные с максимальной скоростью, но для достижения аналогичного показателя при передаче одного фильма следует более тонко настроить приложение. Например, для одного из популярных серверов передачи потокового видео мы обнаружили конфигурационный файл, который позволяет изменять объем буфера передачи данных и размер окна. Поэкспериментировав с этим конфигурационным файлом, мы выявили, что если увеличить размер окна получения (см. далее) в два раза, достигается полная (3,6 Мб/с) скорость передачи одного фильма. Имейте



Рис. 3. Взаимодействие протоколов в режиме тестирования

в виду, что файлы, размещенные в Сети, как правило, сжаты специально для ускорения их загрузки, поэтому для целей тестирования пропускной способности вашей системы они, как правило, не годятся. Для этого нужно более качественное видео, чтобы протестировать систему в действительно стрессовых условиях.

FTP

Продолжим наше тестирование передачей файлов по протоколу FTP (File Transfer Protocol), которая введет дополнительные уровни в виде TCP (Transmission Control Protocol) и сделает нашу мобильную систему передачи данных целостной, соответствующей схеме на рис. 1. Есть три важных параметра, которые следует настроить для достижения максимальной скорости передачи данных по протоколу FTP. Остановимся подробно на каждом: TCP Receive Window, Server Send Buffer и Server Socket Buffer.

Окно получения по TCP (TCP Receive Window)

При передаче файлов по FTP клиент получает данные и использует буфер для того, чтобы процесс передачи происходил непрерывно. Если сервер слишком быстро отправляет данные, буфер клиента наполняется, и он передает серверу команду остановить передачу. Такие процессы (начать — остановить — начать) снижают общую скорость передачи, но этого можно избежать, увеличив размер буфера, называемого окном получения по TCP. Поиск в Интернете по этой фразе дает исчерпывающую информацию и позволяет найти даже бесплатные средства изменения этого параметра для большинства операционных систем. Дело в том, что значение по умолчанию чаще всего слишком невелико для передачи данных с помощью мобильной связи, и его требуется изменить. Мы опытным путем выявили, что для достижения скорости передачи данных в 3,6 Мбит/с достаточно сделать этот параметр равным 64 Кбайт.

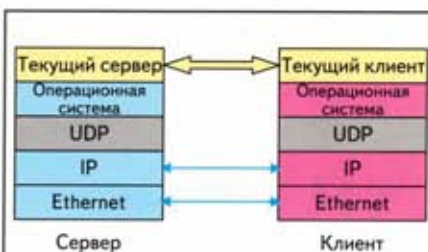


Рис. 5. Система в целом

Буфер отправки сервера (Server Send Buffer)

Большая часть ПО FTP-серверов использует буфер отправки для передачи блоков данных определенного размера вместо непрерывного потока. Если этот параметр невелик, а по умолчанию чаще всего так и есть, системе не удастся достичь максимальной скорости передачи данных. Мы выяснили, что большинство выставляемых значений размера буфера недостаточно для того, чтобы справиться с длительным временем задержки в мобильных системах. Но установка значения этого параметра в 64 Кбайт решает проблему. Размеры буферов отправки и получения можно вычислить, основываясь на том, что иногда называют Bandwidth Delay Product (результатом задержки пропускной способности), но использовать данный метод нужно с осторожностью, так как для нашей системы он не даст корректных значений буферов. Обычно при использовании этого метода используется команда ping для определения задержки передачи данных между клиентом и сервером. В нашей системе нужно использовать команду ping параллельно с передачей данных, а не в то время, когда канал свободен. В мобильных системах время задержки увеличивается по мере роста объема передачи данных, и если применять команду ping во время простоя, вычисленное значение величины буферов будет слишком маленьким. В любом случае 64 Кбайт — достаточно подходящее значение для систем со скоростью передачи данных 3,6 Мбит/с.

Буфер сокетов сервера (Server Socket Buffer)

Большая часть FTP-серверов не дает возможности изменения размера этого буфера, но некото-

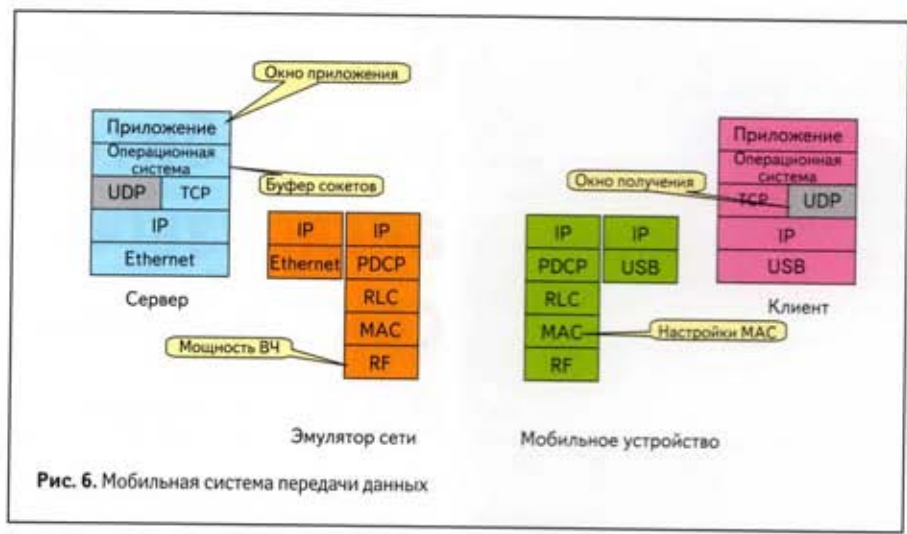


Рис. 6. Мобильная система передачи данных

рые из них все же позволяют это сделать. Размер буфера сокетов напрямую связан с размером буфера отправки, и если его значение недостаточно велико, то буфер отправки быстро переполнит своими данными буфер сокетов, что выльется в снижение скорости передачи. Используйте тот FTP-сервер, который позволяет изменять размер обоих буферов. Установите размер второго буфера чуть больше, чем первого, чаще всего 65 Кбайт вполне достаточно. Имейте в виду, лучше не делать размер буферов слишком большим — это приведет к тому, что некоторые приложения могут останавливаться (а с точки зрения клиента, так и вовсе зависать), потому что будут ожидать полного заполнения буфера перед выполнением передачи между клиентом и сервером. Использование размеров 64 и 65 Кбайт, предлагаемых нами, скорее всего, позволит системе достичь максимальной ско-

рости, добившись которой, можно будет определить пропускную способность тестируемых устройств.

Заключение

На рис. 6 показана использованная нами мобильная система передачи данных с обозначением тех областей, которые требуют особого внимания при настройке перед оценкой результатов работы системы передачи данных. Хотя некоторые производители больше озабочены обеспечением скорости передачи данных лишь на уровне связи мобильного телефона с базовой станцией, конечным пользователям больше интересна скорость передачи данных на общесистемном уровне, поэтому следует убедиться в том, что созданное вами устройство работает не только так, как хотите вы, но и так, как хотят его пользователи.

