Лекция 07а

Протоколы SCTP и DCCP.

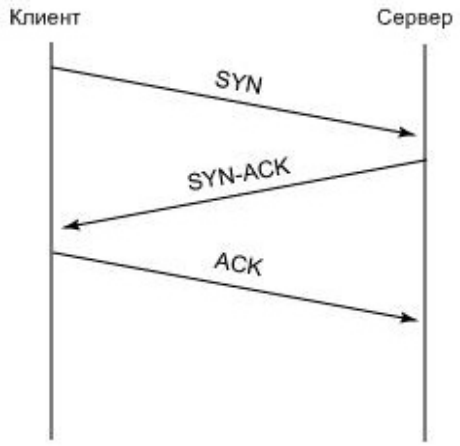
Протокол SCTP

SCTP (Stream Control Transmission Protocol — «Протокол передачи с управлением потоком») — протокол транспортного уровня в компьютерных сетях, появившийся в 2000 году в IETF. **RFC 4960** описывает этот протокол, а RFC 3286 содержит техническое вступление к нему.

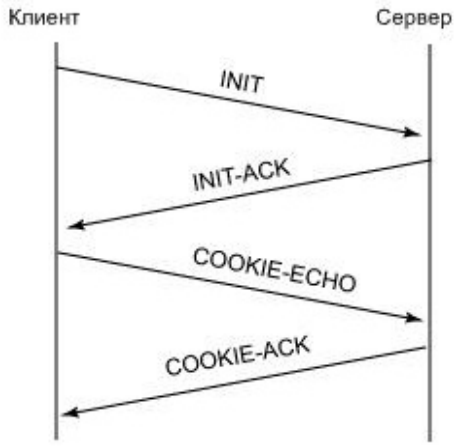
Будучи более новым протоколом относительно TCP, SCTP имеет несколько нововведений, таких как многопоточность, защита от SYN-flood атак, синхронное соединение между двумя хостами по двум и более независимым физическим каналам (multi-homing).

Безопасное установление соединения

Создание нового подключения в протоколах TCP и SCTP происходит при помощи механизма подтверждения (квитирования) пакетов. В протоколе TCP данная процедура получила название трехэтапное квитирование (three-way handshake). Клиент посылает пакет SYN (сокр. Synchronize). Сервер отвечает пакетом SYN-ACK (Synchronize-Acknowledge). Клиент подтверждает прием пакета SYN-ACK пакетом ACK. На этом процедура установления соединения завершается.



Процедура установления соединения (трехэтапное квитирование) в протоколе TCP



Процедура установления соединения (четырехэтапное квитирование) в протоколе SCTP

Протокол TCP имеет потенциальную уязвимость, обусловленную тем, что нарушитель, установив фальшивый IP-адрес отправителя, может послать серверу множество пакетов SYN. При получении пакета SYN сервер выделяет часть своих ресурсов для установления нового соединения. Обработка множества пакетов SYN рано или поздно, затребует все ресурсы сервера и сделает невозможным обработку новых запросов. Такой вид атак называется «отказ в обслуживании» (Denial of Service (DoS)).

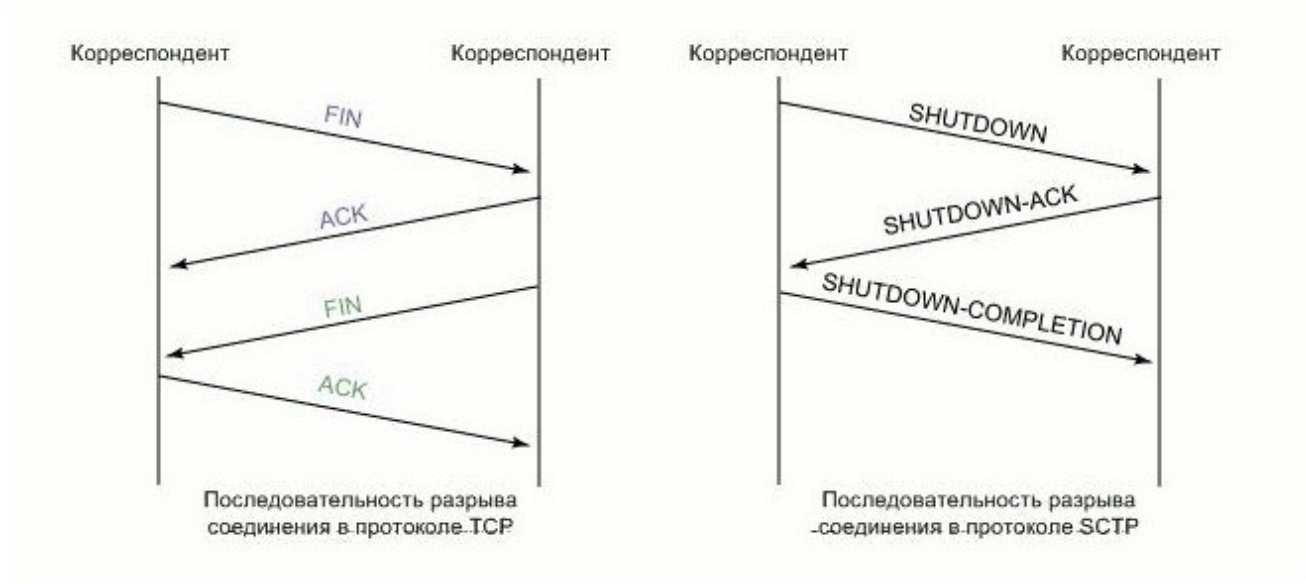
Протокол SCTP защищен от подобных атак с помощью механизма четырёхэтапного

квитирования (four-way handshake) и вводом маркера (cookie). По протоколу SCTP клиент начинает процедуру установления соединения посылкой пакета INIT. В ответ сервер посылает пакет INIT-ACK, который содержит маркер (уникальный ключ, идентифицирующий новое соединение). Затем клиент отвечает посылкой пакета COOKIE-ECHO, в котором содержится маркер, посланный сервером. Только после этого сервер выделяет свои ресурсы новому подключению и подтверждает это отправлением клиенту пакета COOKIE-ACK.

Для решения проблемы задержки пересылки данных при выполнении процедуры четырёхэтапного квитирования в протоколе SCTP допускается включение данных в пакеты COOKIE-ECHO и COOKIE-ACK.

Поэтапное завершение передачи данных

Рассмотрим отличия между процедурой закрытия сокетов протокола SCTP и процедурой частичного закрытия (half-close) протокола TCP.



В протоколе TCP возможна ситуация частичного закрытия соединения, когда один узел закончил передачу данных (выполнив посылку пакета FIN), но продолжает принимать данные по этому соединению. Другой узел может продолжать передавать данные до тех пор, пока сам не проведёт закрытие соединения на своей стороне. Состояние частичного закрытия используется приложениями крайне редко, поэтому разработчики протокола SCTP посчитали нужным заменить его последовательностью сообщений для разрыва существующей ассоциации. Когда узел закрывает свой сокет (посылает сообщение SHUTDOWN), оба корреспондента должны прекратить передачу данных, при этом разрешается лишь обмен пакетами, подтверждающими прием ранее отправленных данных.

Многопоточность

TCP управляет последовательностью байт: данные, посланные приложением-отправителем, должны поступать приложению-получателю строго в том же порядке (в то время как протокол IP способен поменять последовательность пакетов; кроме того, пропавшие пакеты посылаются повторно и обычно прибывают к получателю с нарушением последовательности; для борьбы с этими явлениями данные накапливаются в буфере). SCTP может транспортировать данные между двумя точками (узлами) одновременно по нескольким потокам сообщений. В противоположность к TCP, SCTP обрабатывает целые сообщения, а не обычные байты информации. Это означает, что если отправитель отсылает серверу сообщение, состоящее из 100 байт за первый шаг, а за ним ещё 50 байт, то получатель за первый шаг получит именно первые 100 байт в первом сообщении, а только затем 50 байт на второй операции чтения из сокета.

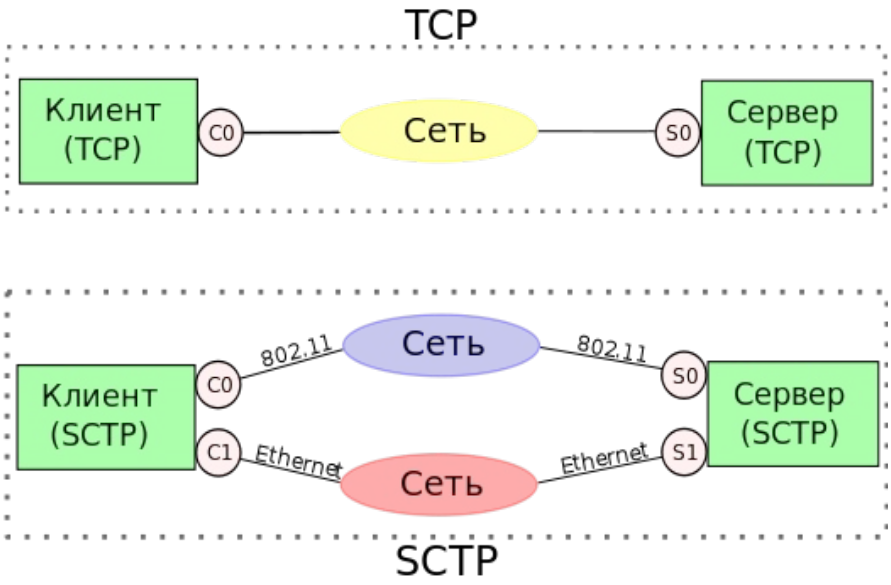
Термин «многопоточность» (англ. multi-streaming) обозначает способность SCTP параллельно передавать по нескольким независимым потокам сообщений. Например, мы передаем несколько фотографий через HTTP-приложение (например, браузер). Можно использовать для этого связку из нескольких TCP-соединений, однако также допустимо SCTP-ассоциация (англ. SCTP-association), управляющее несколькими потоками сообщений для этой цели.

TCP достигает правильного порядка байт в потоке, абстрактно назначая порядковый номер каждой отосланной единице, а упорядочивая принятые байты, используя назначенные порядковые номера, по мере их прибывания. С другой стороны, SCTP присваивает различные порядковые номера сообщениям, посылаемым в конкретном потоке. Это разрешает независимое упорядочивание сообщений по разным потокам. Так или иначе, многопоточность является опцией в SCTP. В зависимости от желаний пользовательского приложения, сообщения могут быть обработаны не в порядке их отправления, а в порядке их поступления.

Достоинства использования SCTP:

Использование множественных интерфейсов (Multihoming)

Допустим, у нас есть два хоста. И хотя бы один из них имеет несколько сетевых интерфейсов, и соответственно несколько IP-адресов. В TCP, понятие «соединение» означает обмен данными между двумя точками, в то время, как в SCTP имеет место концепция «ассоциации» (англ. association), обозначащая всё происходящее между двумя хостами



Многопоточность

Данные приходят в точку по независимым потокам.

Поиск пути с мониторингом

Протоколом выбирается первичный маршрут передачи данных, а также производится проверка и мониторинг связности пути.

Механизмы валидации и проверки подлинности

Защита адресата от flood-атак (технология 4-way handshake), и уведомление о потерянных пакетах и нарушенных цепочках.

Улучшенная система контроля ошибок, подходящая для jumbo-пакетов в Ethernet.

Протокол DSSP

DCCP (Datagram Congestion Control Protocol) — протокол транспортного уровня модели OSI, разрабатываемый IETF. Принят в качестве стандарта в марте 2006 года. Он предоставляет механизмы для отслеживания перегрузок в сети, избегая необходимости создавать их на прикладном уровне. Этот протокол не гарантирует доставку информации в нужном порядке.

DCCP очень эффективен для приложений в которых данные пришедшие не вовремя становятся бесполезными. Например: потоковое медиа-вещание, онлайн игры и интернет-телефония. Главная особенность этих приложений состоит в том, что старые сообщения очень быстро становятся бесполезными, поэтому лучше получить новое сообщение, чем пытаться переслать старое. Но на данный момент большинство таких приложений самостоятельно реализовывают отслеживание перегрузок, а в качестве протокола передачи используются TCP или UDP.