



Управление перегрузкой: TCP Reno, TCP New Reno

Введение в компьютерные сети
проф. Смелянский Р.Л.
Лаборатория Вычислительных комплексов
ф-т ВМК МГУ



TCP Tahoe vs TCP Reno

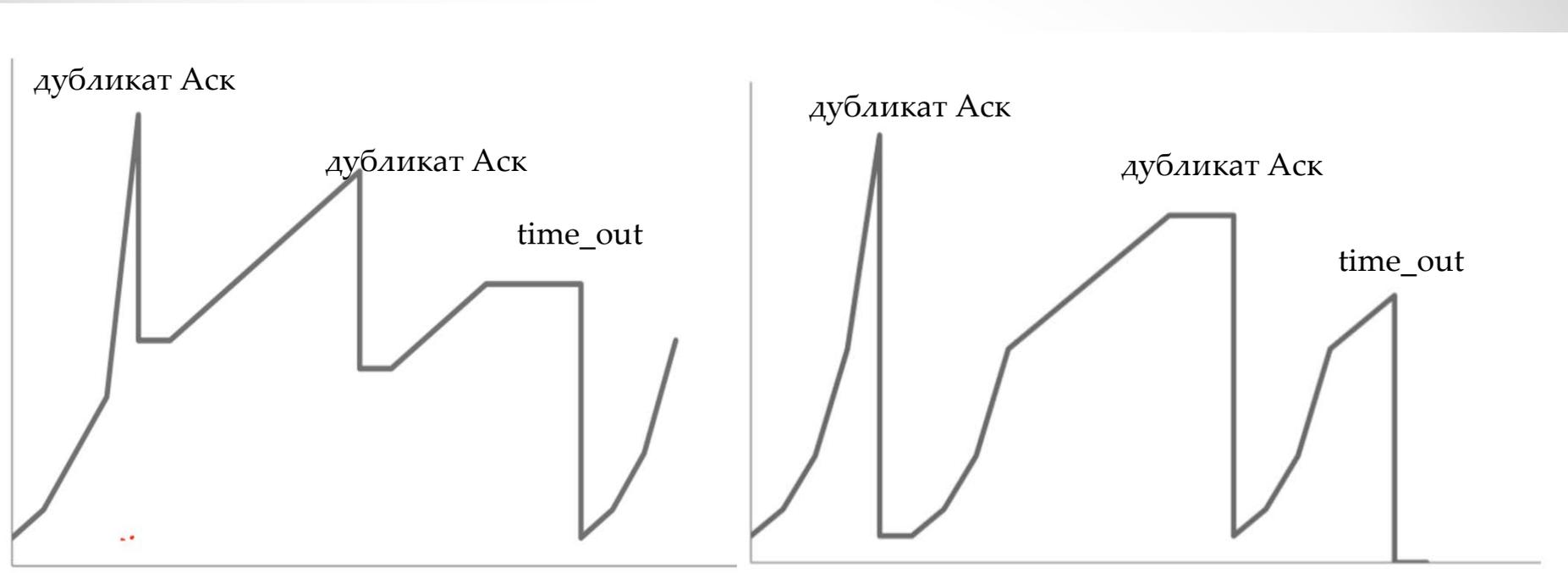
- *TCP Tahoe* - Не ждать *time-out*, а уже по тройному уведомлению (подразумевается потеря пакета) начинать повторную пересылку и переход в фазу медленного старта
 - установить порог = $CWND/2$
 - сбросить $CWND$ в 1
 - начать медленный старт
- *TCP Reno* - быстрое восстановление:
 - по тройному ACK $cwnd = cwnd/2$, а не 1
 - на каждый дубль ACK $cwnd = cwnd+1$
 - шлем запрашиваемый сегмент после 3-го дубля



TCP Reno

TCP Tahoe

Р
а
з
м
е
р
о
к
н
а



Reno

Tahoe

время



TCP Reno: пример

receiver



sender



Time

TCP New Reno

- По *time-out* поведение такое же как и у Tahoe/Reno
- В фазе быстрого восстановления:
 - при входе в эту фазу - запомнить последний не подтвержденный пакет
 - при каждом повторном уведомлении - увеличить *CWND* на *MSS*
 - Начать отправку новых пакетов пока находимся в фазе быстрого восстановления
 - Когда последний пакет подтвержден:
 - вернуться в фазу избегания перегрузки
 - восстановить размер *CWND* до того размера, который оно имело до входа в фазу быстрого восстановления

Управление перегрузками

- *Одна из сложнейших проблем в компьютерных сетях (особенно на неоднородных, протяженных, с ошибками соединениях)*
- *Основной подход: AIMD (additive increase, multiple decrease)*
- *Для того чтобы поддерживать канал заполненным в соответствии с его пропускной способностью, хорошо бы:*
 - *Быстрое повторная пересылка данных (не дожидаясь `time_out`)*
 - *Увеличение `CWDN` (не ждать `RTT`, чтобы послать новые данные)*
- *Дома построить диаграмму состояний для TCP Reno*