

54 Усовершенствования протокола ТСП

Следует отметить, что базовая редакция стандарта ТСП предоставила реализациям определенную вольность (как во многих других случаях со стандартами). Это привело к тому, что при полном соблюдении требований во многих случаях оконный механизм ТСП может оказаться неэффективным. «Разношерстность» реализаций усугубляет ситуацию.

Как следствие, потребовалась разработка дополнений к базовому алгоритму. Новые алгоритмы оперируют с новыми понятиями.

В частности, хорошо известна проблема, вошедшая в историю под обобщенным названием «синдром глупого окна» («silly window syndrome»), в свое время «стопорившая» значительную часть пространства Internet.

Синдром может возникать по разным причинам и проявляется в том, что текущее окно передачи не соответствует состоянию приемника, тем самым не позволяя его как следует «нагрузить» либо, наоборот, «разгрузить».

Решение Нэгла (Nagle) позволяет побороть «синдром глупого окна» когда передающей стороне требуется часто отправлять небольшие сегменты с данными.

Решение Кларка (Clark) позволяет побороть «синдром глупого окна» когда принимающей стороной часто анонсируется небольшое предлагаемое окно.

Также стандартизированы четыре дополнения Ван Якобсона (Van Jacobson), призванные бороться с перегрузками в СПД (последнее RFC -- RFC 5681):

1. Медленный старт (slow start).

Идея заключается в том, что в начале передачи размер текущего окна передачи нужно увеличивать не «скачком», а плавно, пропорционально скорости получения подтверждений (не превышая размер предлагаемого окна).

Рекомендуемые формулы:

$IW = 2 * SMSS$, если $SMSS > 2190$ Bytes ,
 $IW = 3 * SMSS$, если $2190 \text{ Bytes} \geq SMSS > 1095$ Bytes ,
 $IW = 4 * SMSS$, если $SMSS \leq 1095$ Bytes ,

где IW (initial window) -- начальное значение текущего окна передачи:

$cwnd += \min (N, SMSS)$,

где $cwnd$ (congestion window) -- текущее окно передачи (в данном случае, окно затора), N -- количество подтвержденных байтов, $SMSS$ (sender MSS) -- MSS передатчика.

2. Избегание затора (congestion avoidance).

Состоит в сдерживании экспоненциального роста размера текущего окна передачи после преодоления им некоторого порога. Как правило переход к избеганию затора происходит после медленного старта.

Рекомендуемые формулы:

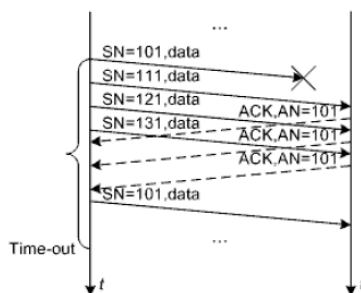
$ssthresh = \max (FlightSize / 2, 2 * SMSS)$,

где $ssthresh$ (slow start threshold) -- порог перехода от медленного старта к избеганию затора, $FlightSize$ -- количество еще неподтвержденных байтов;

$cwnd += SMSS * SMSS / cwnd$.

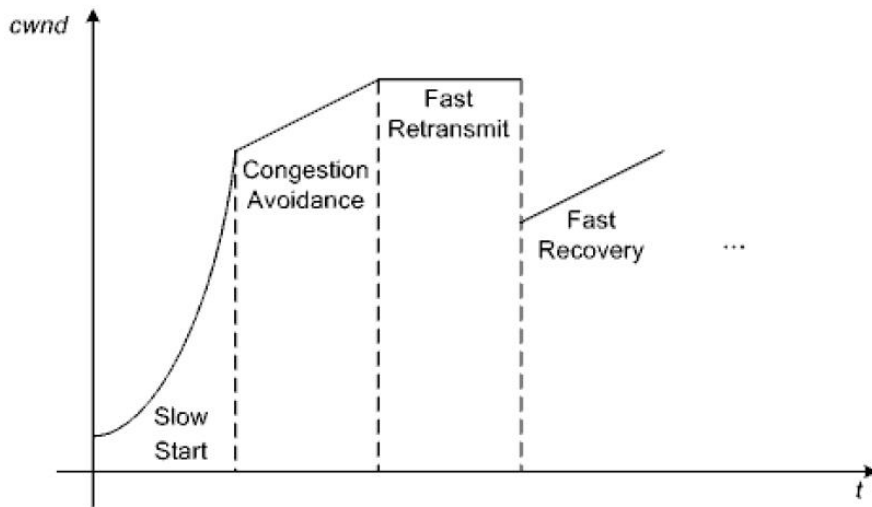
3. Быстрая повторная передача (fast retransmit).

При получении принимающей стороной разупорядоченного сегмента с данными (возможно из-за потери ожидаемого сегмента с данными) незамедлительный повтор подтверждения с AN недостающего сегмента с данными. При получении передающей стороной трех одинаковых подтверждений незамедлительный повтор сегмента с данными согласно AN. Что, в некоторых ситуациях, позволяет успешно переслать потерянный сегмент еще до наступления тайм-аута.



4. Быстрое восстановление (fast recovery).

После обнаружения затора, переход сразу к избеганию коллизий, минуя стадию медленного старта. Как правило в связке с быстрой повторной передачей.



TCP congestion control

Последствия потерь и разупорядочивания сегментов заключаются в разрушении «маятника» взаимодействия и приводят к необходимости еще одной важной оптимизации, четко проявляющейся при быстрой повторной передаче.

Согласно базовому алгоритму все сегменты должны быть подтверждены, а значит, после быстрой повторной передачи принимающая сторона должна послать все недостающие подтверждения.

Но стороны могут «договориться», что текущий AN отражает номер первого ожидаемого получателем сегмента плюс автоматически подтверждает все сегменты с меньшими номерами (cumulative acknowledgement).