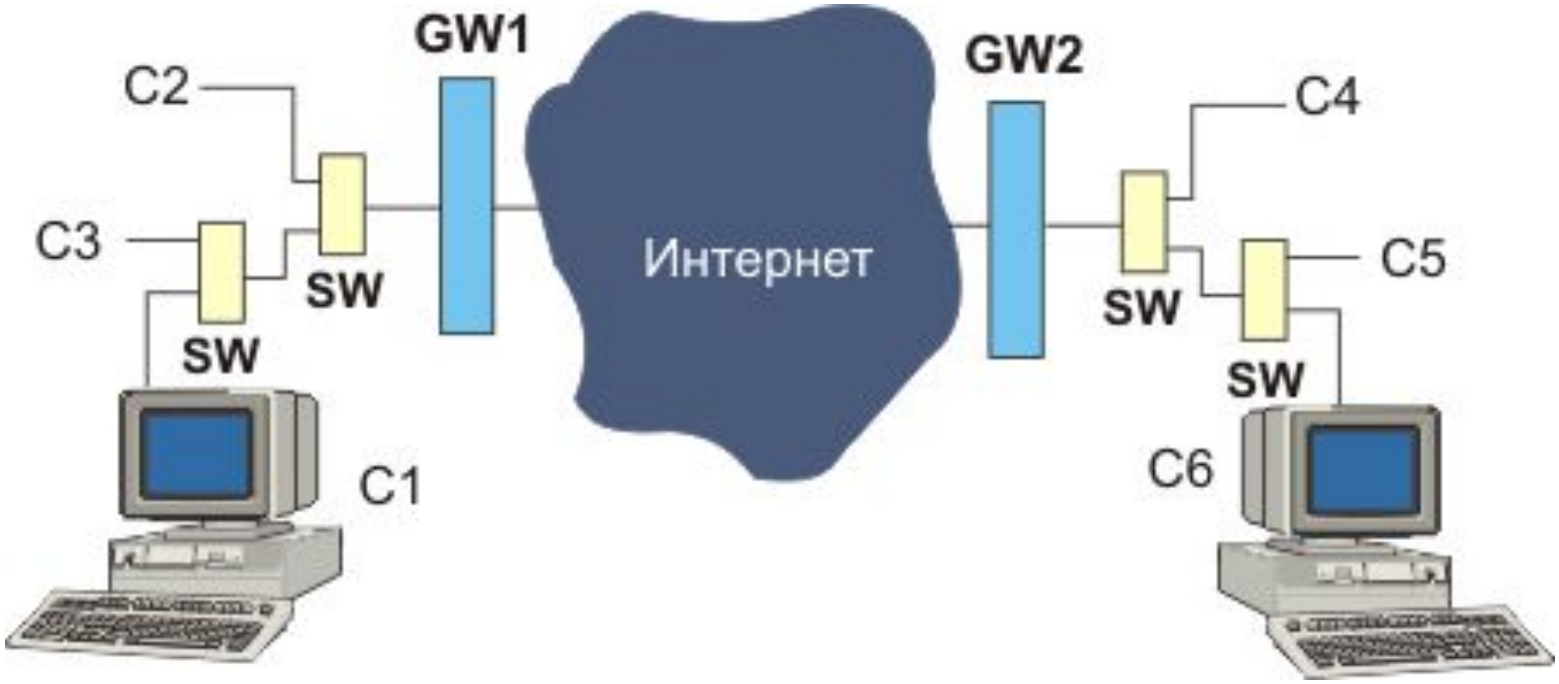
Модели TCP



- 1. Вероятность ошибки доставки (BER) невелика и потеря пакета вероятнее всего происходит из-за переполнения буфера. Если потеря пакета из-за его искажения существенна, понижение CWND не поможет, и пакеты будут теряться с той же вероятностью (здесь было бы уместно поискать оптимальное значение MTU).
- 2. Время доставки (RTT) достаточно стабильно и для его оценки можно использовать простые линейные аппроксимации. Здесь подразумевается, что в рамках сессии все пакеты следуют одним и тем же путем и смена порядка прихода пакетов, хотя и допускается, но маловероятна. Разрешающая способность внутренних часов отправителя должна быть достаточно высока, в противном случае возникают серьезные потери из-за таймаутов.
- 3. Сеть имеет фиксированную полосу пропускания и, во всяком случае, не допускает скачкообразных ее вариаций. В противном случае потребовался бы механизм для прогнозирования полосы пропускания, а действующие алгоритмы задания CWND оказались бы не эффективными
- 4. Буферы сетевых устройств используют схему первый_вошел-первым_вышел (FIFO). Предполагается, что размер этих буферов соответствует произведению RTT*B (B - полоса пропускания, RTT - сумма времен транспортировки сегмента от отправителя к получателю и времени движения отклика от получателя к отправителю). Если последнее условие нарушено, пропускная способность неизбежно

- 5. Длительность TCP-сессии больше нескольких RTT, чтобы оправдать используемую протокольную избыточность. Короткие TCP-сессии, широко используемые WEB-технологией снижают эффективность обмена. (Именно это обстоятельство вынудило в версиях HTTPv1.1 и выше не разрывать TCP-соединение после вызова очередной страницы).
- 6. Чтобы минимизировать влияние избыточности, связанной с заголовком (20 байт IP +20 байт TCP + MAC-заголовок), используемое поле данных должно иметь большой объем. Для узкополосных каналов, где MTU мало, нарушение данного требования делает канал низкоэффективным. По этой причине выявление допустимого MTU в начале сессии должно приветствоваться.
- 7. Взаимодействие с другими TCP-сессиями не должно быть разрушительным, приводящим к резкому снижению эффективности виртуального канала

- Данные условия выполняются отнюдь не всегда, и система не рухнет, если эти условия нарушаются часто. Но эффективность работы соединения окажется не оптимальной.

- ***Трудности в реализации модели протокола TCP возникли при работе с современными быстрыми (1-10 Гбит/с) и длинными (RTT>200мсек) каналами. Для пакетов с длиной 1500 байт время формирования окна оптимального размера достигает 83333 RTT (режим предотвращения перегрузки), что при RTT=100мсек составляет 1,5 часа!***

TCP-reno

$$cwnd(t+t_A) = \begin{cases} \text{фаза медленного старта:} \\ cwnd(t)+1, \quad if \quad cwnd(t) < ssth(t); \\ \text{фаза исключения перегрузки} \\ cwnd(t) + \dfrac{1}{cwnd(t)}, \quad if \quad cwnd(t) \geq ssth(t); \end{cases}$$

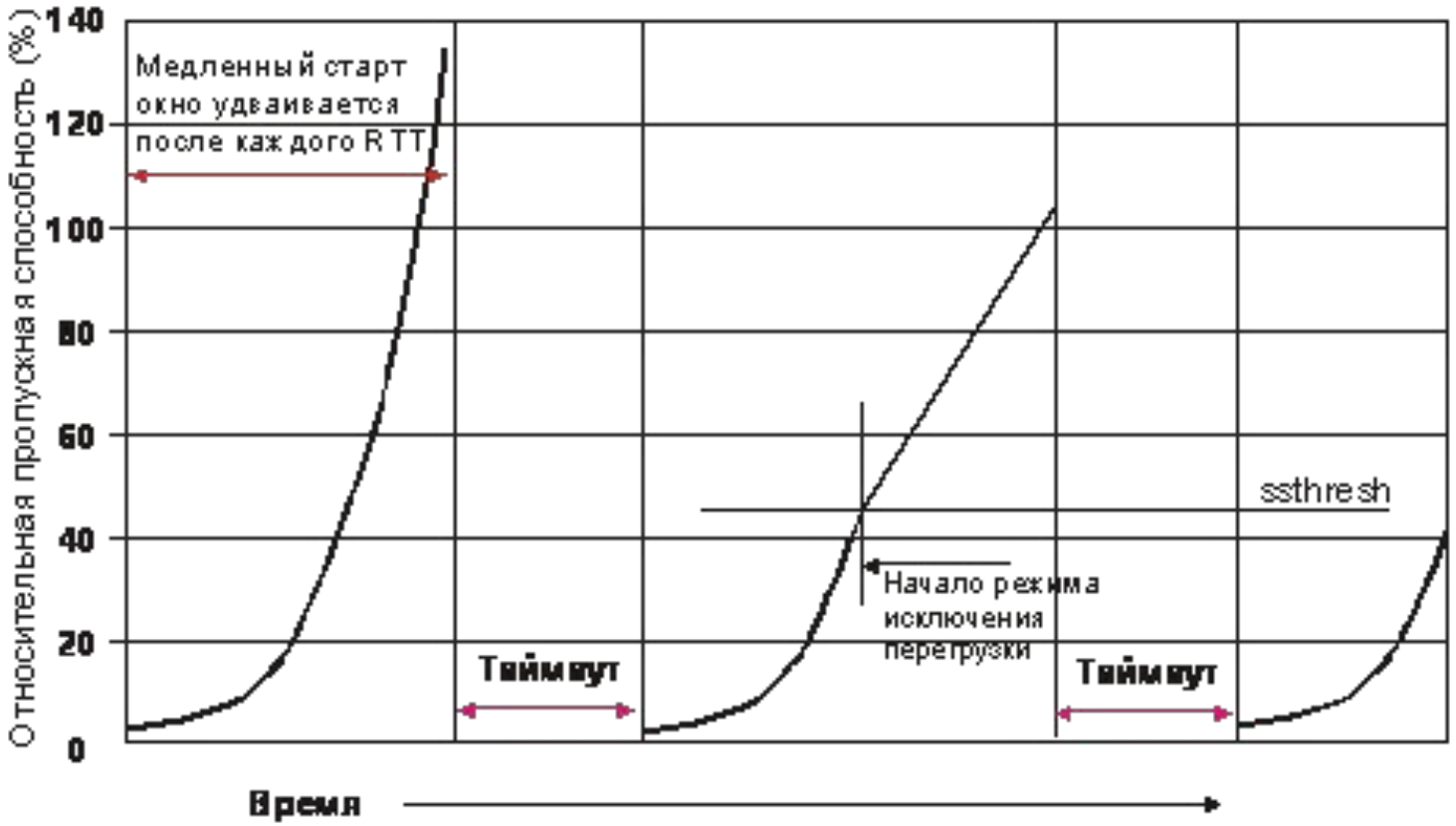
cwnd(t)=1; ssth(t)=(cwnd(t))/2;

В настоящее время наиболее популярной является модель **NewReno**, изложенная в документе RFC-3782 (апрель 2004 года), и использующая алгоритм Fast Retransmit & Fast Recovery. Алгоритм NewReno использует переменную **recover** (восстановление), исходное значение которой равно исходному порядковому номеру пакета.

TCP Vegas

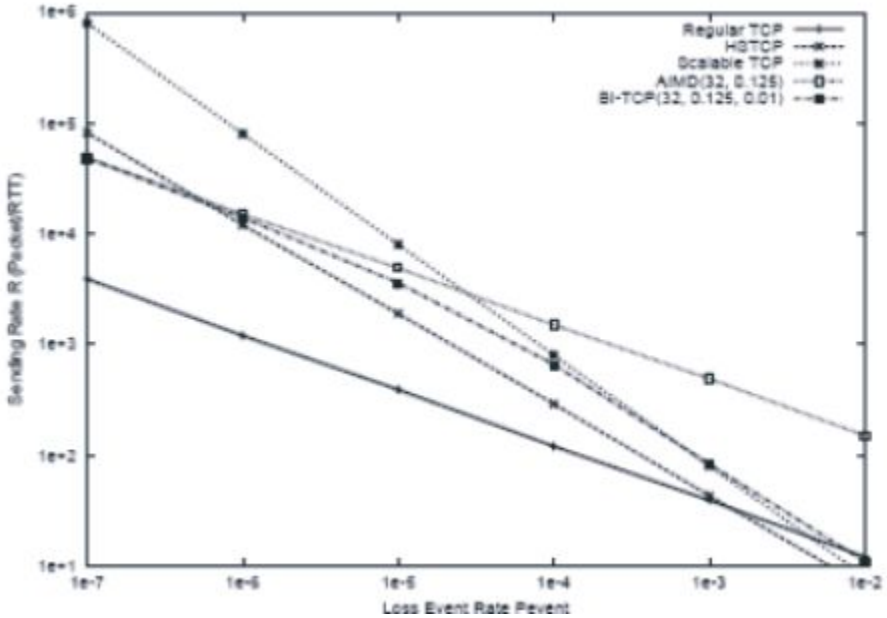
$$cwnd(t+t_A) = \begin{cases} cwnd(t)+1, & if \quad diff < \dfrac{\alpha}{base\_rtt} \\ cwnd(t), & \dfrac{\alpha}{base\_rtt} \le diff \le \dfrac{\beta}{base\_rtt}; \quad diff = \dfrac{cwnd(t)}{base\_rtt} - \dfrac{cwnd(t)}{rtt} \\ cwnd(t)-1, & \dfrac{\beta}{base\_rtt} < diff \end{cases}$$

TCP-Tahoe



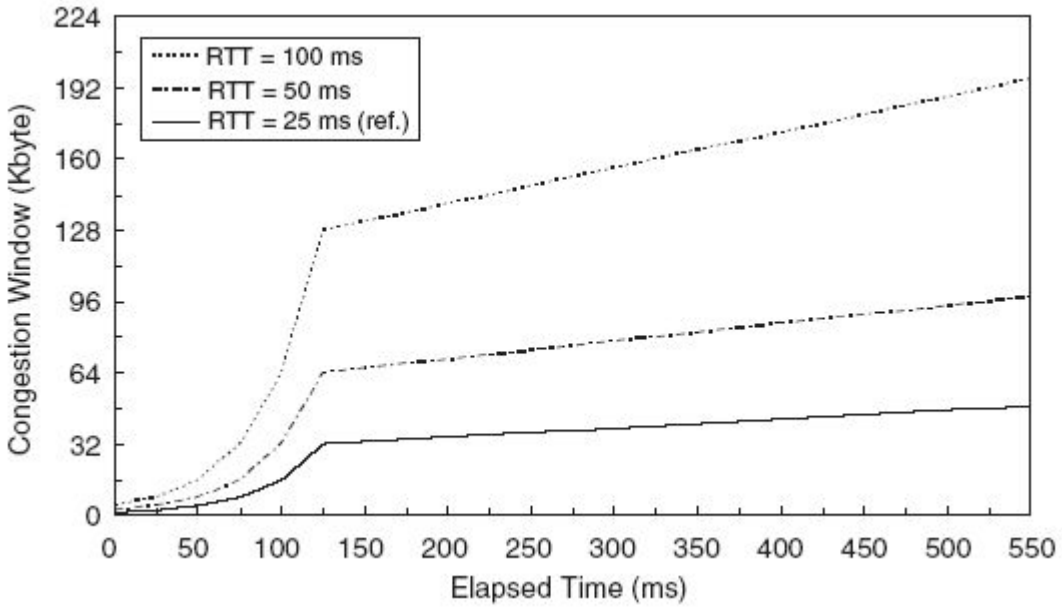
$$RTO = \overline{RTT} + 4\sigma_{\overline{RTT}}$$

Сравнение функций отклика для разных протоколов

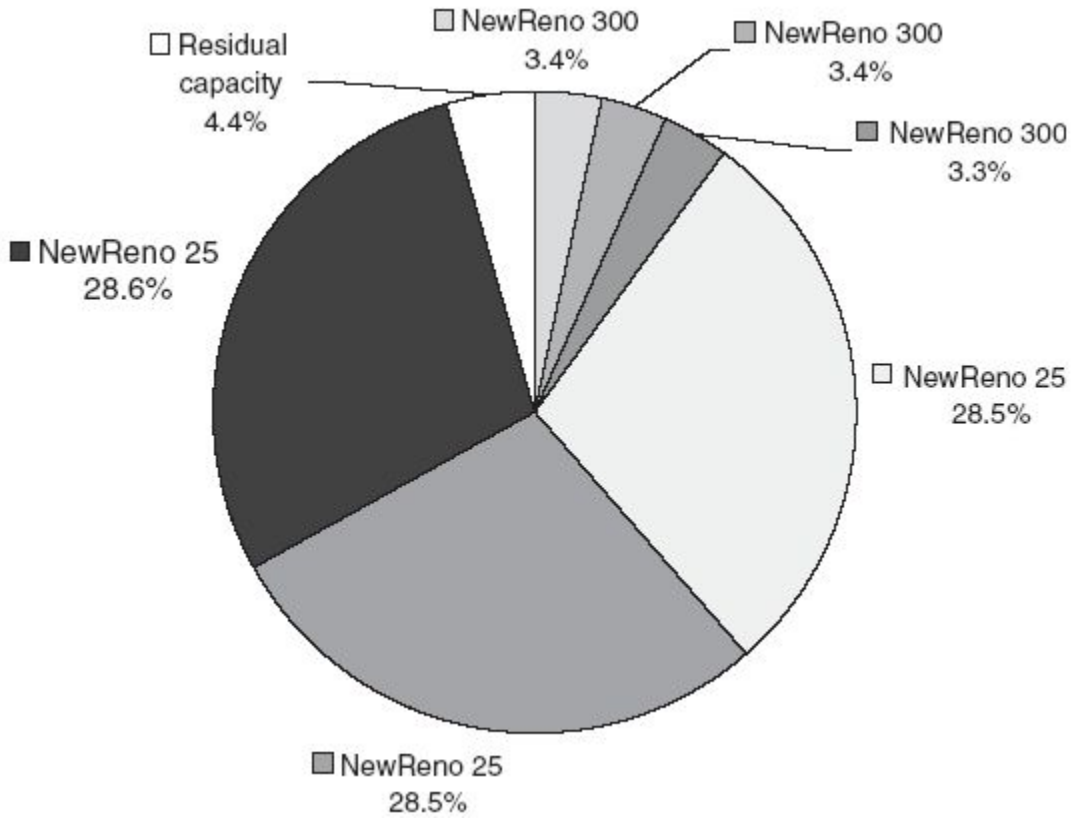


Алгоритм TCP HYBLA

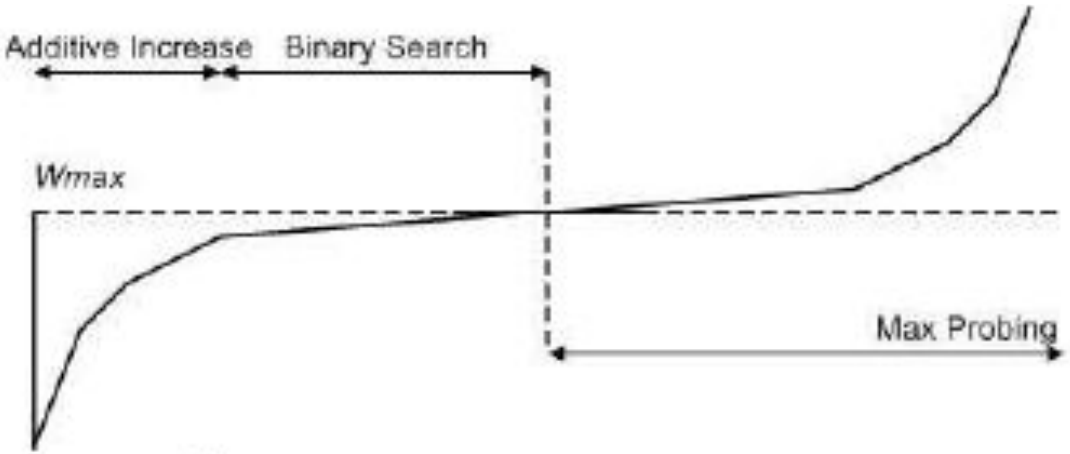
Основной идеей TCP Hybla является достижение для соединений с большим (напр. спутниковых) тех же скоростей передачи, B(t), что и для проводных TCP-каналов



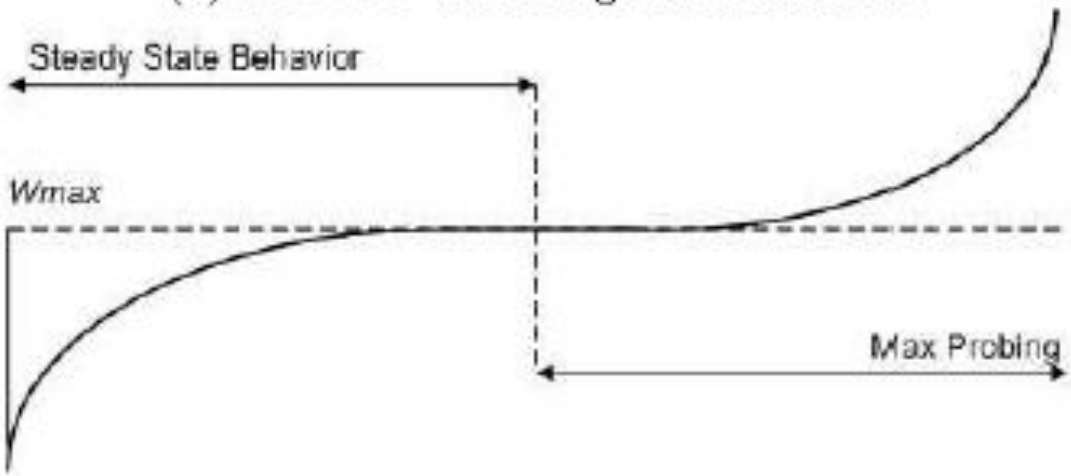
Несовершенство версии протокола TCP Newreno для каналов с разными значениями RTT



CUBIC



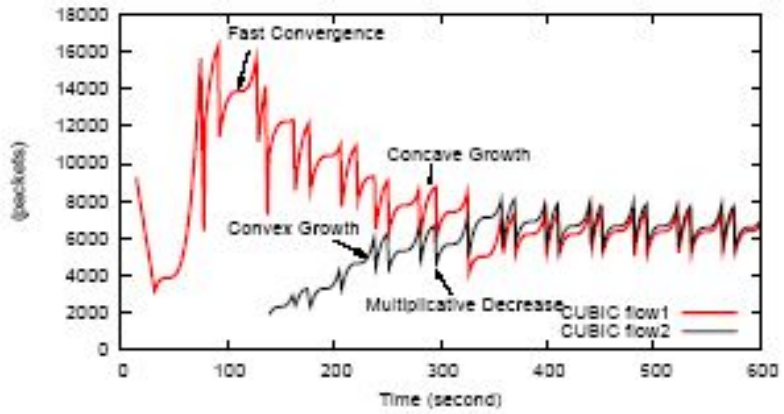
(a) BIC-TCP window growth function.



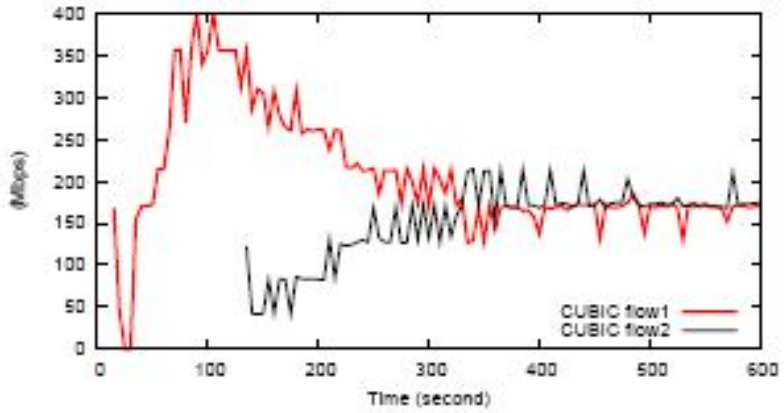
(b) CUBIC window growth function.

- Рост окна в модели CUBIC осуществляется в соответствии с выражением:
- $W(t) = C(t-K)^3 + W_{max}$
- где C параметр CUBIC, t - время с момента последнего уменьшения ширины окна, а K равно периоду времени, который необходим для увеличения W до W_{max}, его значение вычисляется с привлечением выражения:
- $$K = \sqrt[3]{\frac{W_{max}\beta}{C}}$$

Two CUBIC flows with 246ms RTT



(a) CUBIC window curves.



(b) Throughput of two CUBIC flows.

Работа протокола TCP AIMD

- **A**dditive-**I**ncrease, **M**ultiplicative-**D**ecrease (Область линейного увеличения CWND)
- Работа протокола TCP AIMD в режиме исключения перегрузок можно характеризовать формулой:
- BW= $\frac{MSS}{(RTT \times \sqrt{(1.33 \times \rho)}) + (RTO \times \rho \times [1 + 32 \times \rho^2] \times \min(1, 3 \times \sqrt{0.75 \times \rho}))}$
- где BW - полоса пропускания;
- MSS - максимальный размер сегмента в байтах, используемый сессией.
- RTO - таймаут повторной пересылки.
- **ρ** - частота потери пакетов (0.01 означает 1% потерь)
- Эта формула является наилучшей аппроксимацией. Некоторое упрощение формулы можно получить, считая RTO=5*RTT.
- Более упрощенная формула $BW = 0.93 \times \frac{MSS}{RTT\sqrt{\rho}}$

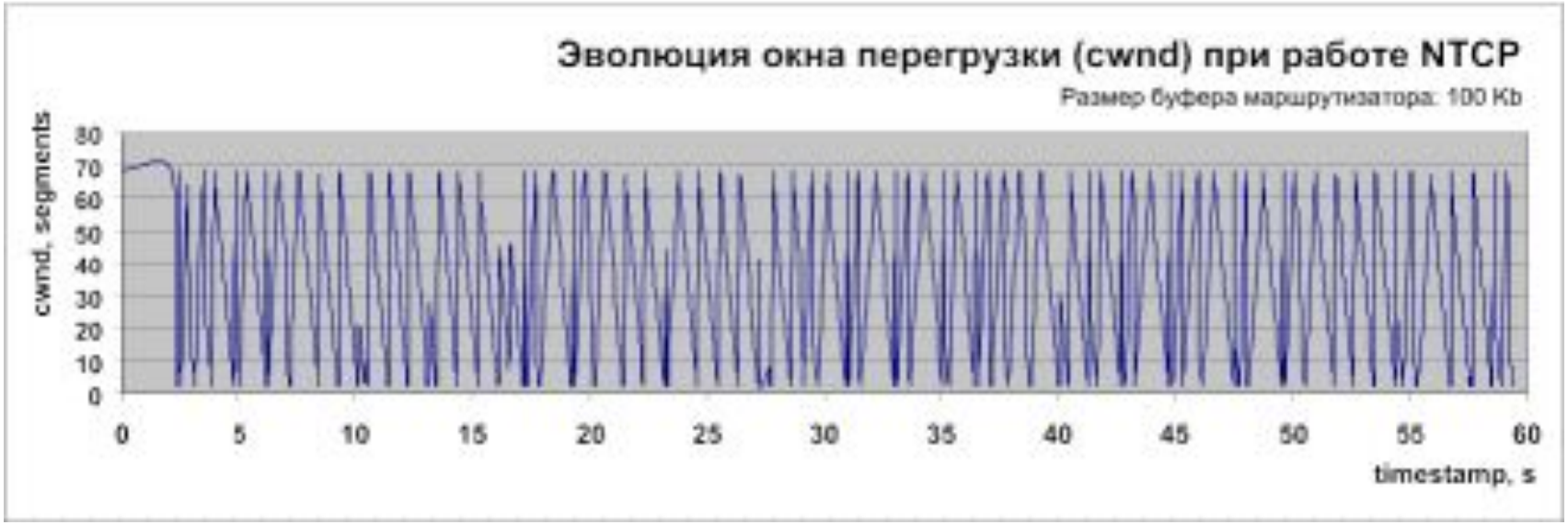
Взаимодействие с чужими потоками

- При получении трех дублированных подтверждений (DUPACK) отправитель считает пакет потерянным и посылает его повторно.
- каждое соединение обычно теряет около двух пакетов в каждом эпизоде перегрузки ***В среднем следует ожидать потерю трех пакетов на одно столкновение***.
- **ECN** - Explicit Congestion Notification

NTCP

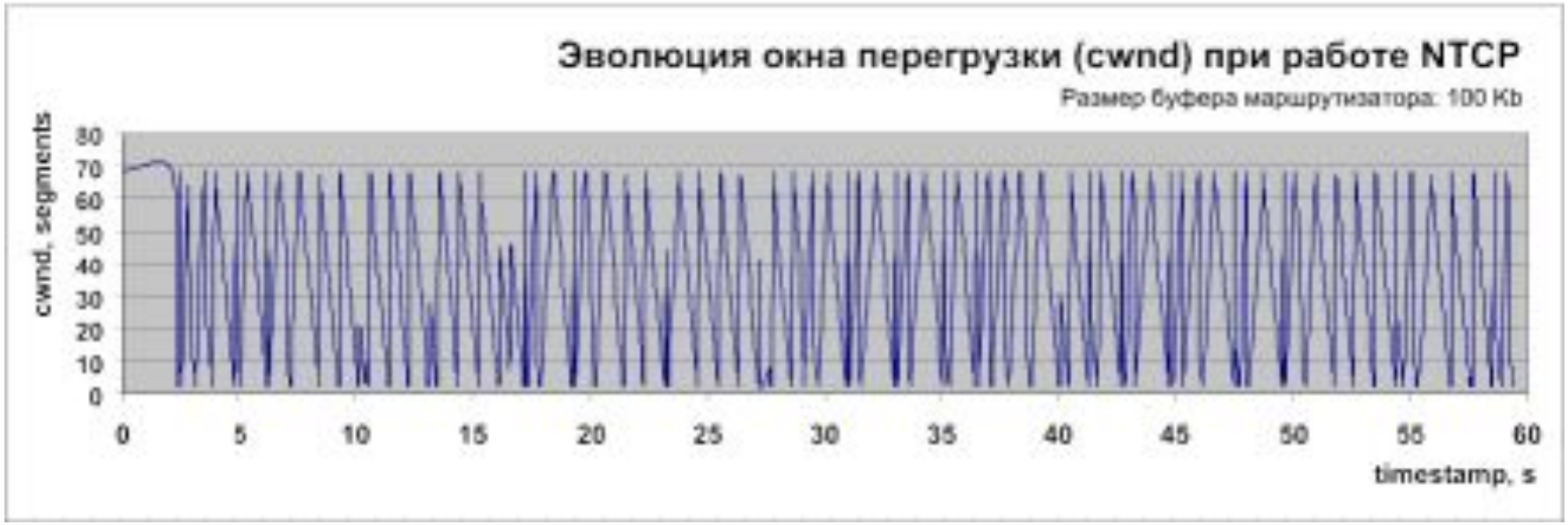
- Темп заполнения буфера определяется производной db/dt. Если уровень заполнения достигает $\mathbf{B_{max}}$, следующий пришедший сегмент будет потерян. Значение $\mathbf{B_{max}}$ в общем случае определяется неравенством $\mathbf{B_{max}} > B \times RTT/MSS$. Сетевое устройство должно отслеживать уровень заполнения своего буфера. И, если после получения очередного сегмента оказывается, что
- $\mathbf{(b(t) + db/dt \times RTT + \delta) > B_{max}}$,
- то всем отправителям-соседям, которые используют данное устройства для передачи данных, должен быть послан отклик с window=0 (сигнал прекращения передачи). δ - конфигурационный параметр.

NTCP



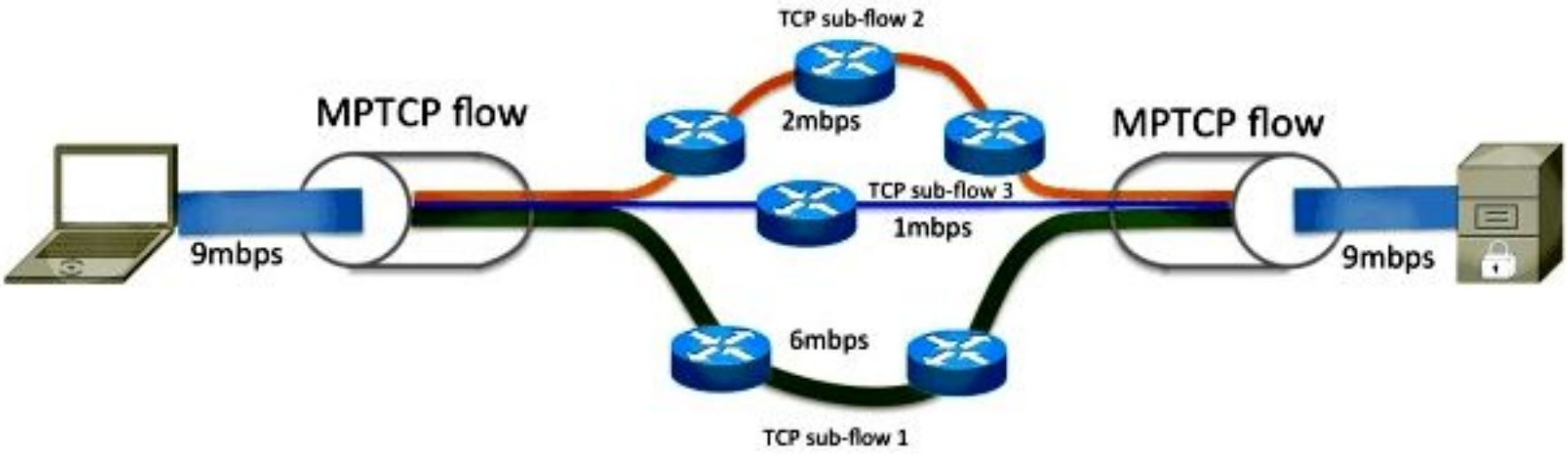
То же, что и на предыдущем рисунке но для протокола NTCP. Здесь протокол, предвидя переполнение буфера, реагирует снижением CWND

NTCP


Эволюция окна перегрузки (cwnd) при работе NTCP
Размер буфера маршрутизатора: 100 Kb
cwnd, segments
80
70
60
50
40
30
20
10
0
0
5
10
15
20
25
30
35
40
45
50
55
60
timestamp, s

Multipath TCP

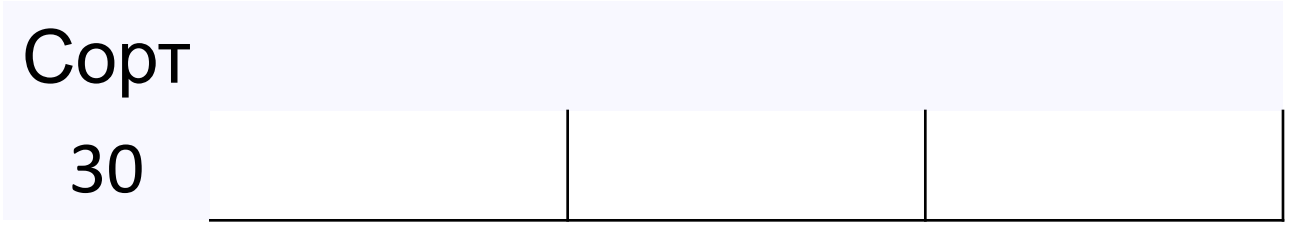
RFC-6824 TCP Extensions for Multipath Operation with Multiple Addresses



Сравнение стандартного TCP и стеков MPTCP-протокола

Пример сценария использования MPTCP

Формат опций MPTCP

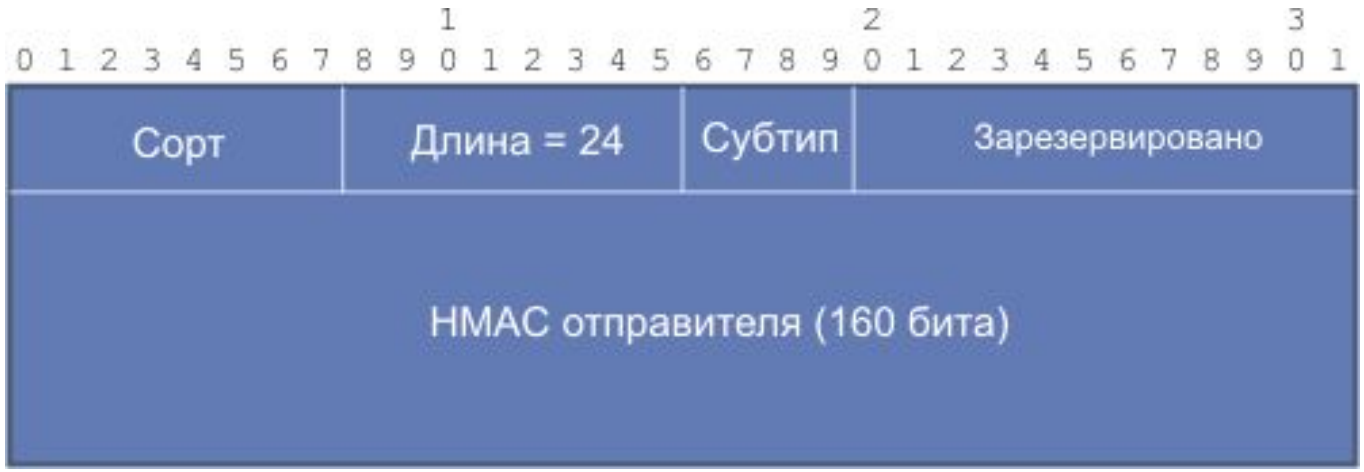


Опция MP_CAPABLE

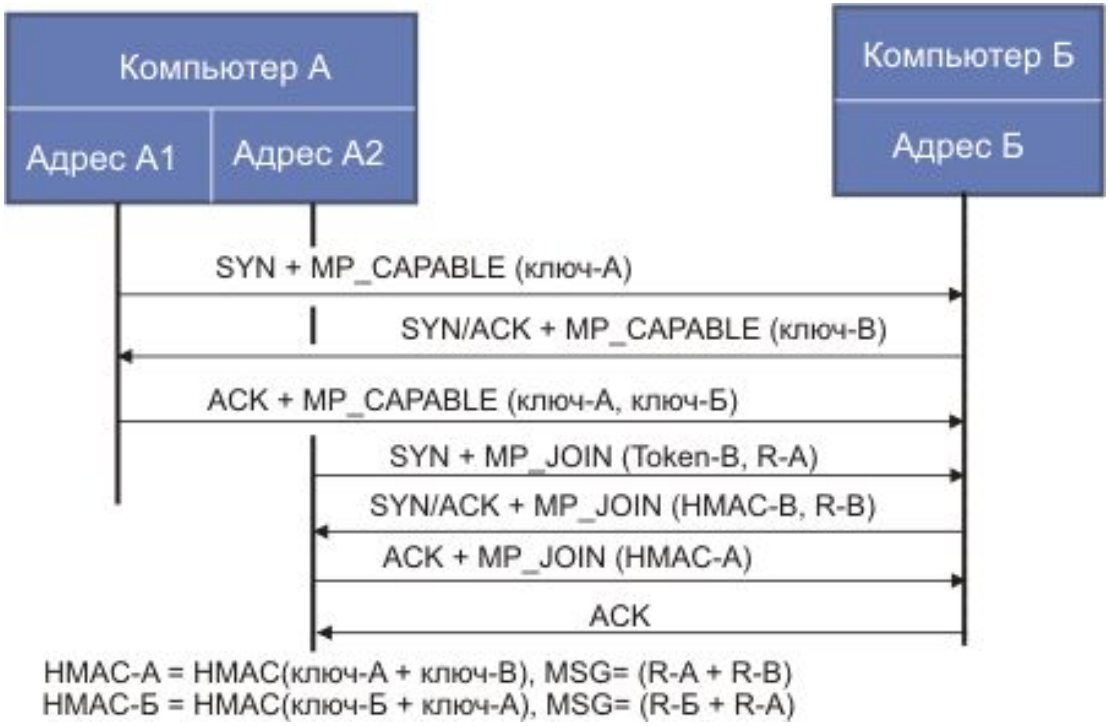
A=1 = Необходима контрольная сумма
B=0, является флагом расширения
с C по H - используются для согласования используемого криптоалгоритма.

Опция MP_JOIN (для исходного SYN)

Опция Join соединение (MP_JOIN) (для третьего ACK)



Пример использования аутентификации в MPTCP



Опция DSS (Data Sequence Signal)

A = Data ACK присутствует
a = Data ACK имеет 8 октетов (если a=0, Data ACK имеет 4 октета)
M = **DSN** (Data Sequence Number - порядковый номер данных]), **SSN** (Subflow Sequence Number - порядковый номер субпотока), длина уровня данных и контрольная сумма присутствуют.
m = порядковый номер данных имеет 8 октетов (если не определен, DSN имеет 4 октета)