

Transmission Control Protocol (TCP)

TCP: Услуги

- RFCs 793, 1122, 1323, 2018, 2581, 2873, 2988, 3168, 4614
- **Надеждна комуникация между два приложни процеса (приложения)**
 - Гарантира, че данните ще бъдат доставени без загуба, дублирания или грешки.
 - През множество надеждни и/или ненадеждни мрежи
 - С контролиране на потока, грешките и задръстванията
- **С използване на съединение**
 - Приложението изисква съединение за трансфер на данните
 - **Установяване на надеждно съединение**
 - 3-кратно ръкостискане гарантира надежден и синхронизиран старт на комуникацията между двете крайни точки
 - Съединението се установява между двойка *сокети*
 - *Сокет* = IP адрес + номер на порт
 - 1 *сокет* може да се използва за няколко съединения едновременно
 - **Разпадане на съединението**
 - Гарантиране доставката на всички данни преди разпадане на съединението
 - 4(3)-кратно ръкостискане
- **От точка до точка (E2E)**
 - Съединението има 2 крайни точки, идентифицирани чрез *сокети*.
 - Няма поддръжка на *multicasting* и *broadcasting*!

TCP: Услуги (прод.)

- **Пълн дуплекс**
 - Крайните точки обменят данни в двете посоки едновременно
 - Всяка точка разполага с буфер за изпращане и получаване на данни
- **Байтово-ориентирани**
 - Приложният процес предава своето съобщение към TCP
 - TCP го разглежда като непрекъснат поток от байтове
 - TCP може да **не** запази границите на съобщението
 - Може да изпрати съобщението веднага, или
 - Може да го буферира с цел акумулиране на по-голям обем от данни за изпращане наведнъж

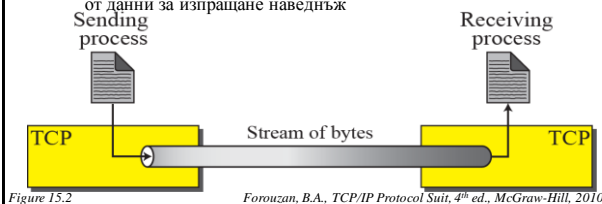


Figure 15.2

Forouzan, B.A., TCP/IP Protocol Suit, 4th ed., McGraw-Hill, 2010

TCP: Използване на портове

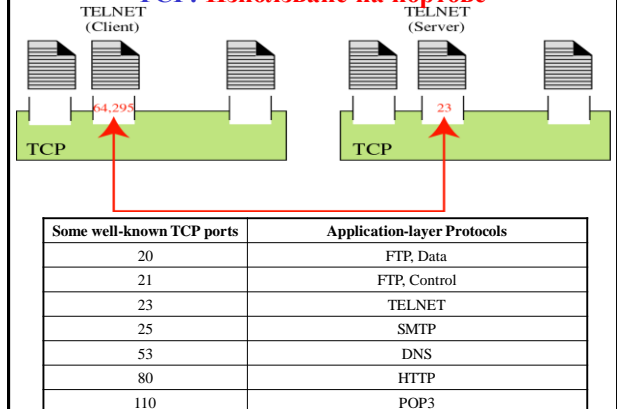


Figure 12-3

Forouzan, B.A., TCP/IP Protocol Suit, 2nd ed., McGraw-Hill, 2003

TCP: Трансфер на данни



Figure 15.3

Forouzan, B.A., TCP/IP Protocol Suit, 4th ed., McGraw-Hill, 2010

TCP: Сегменти

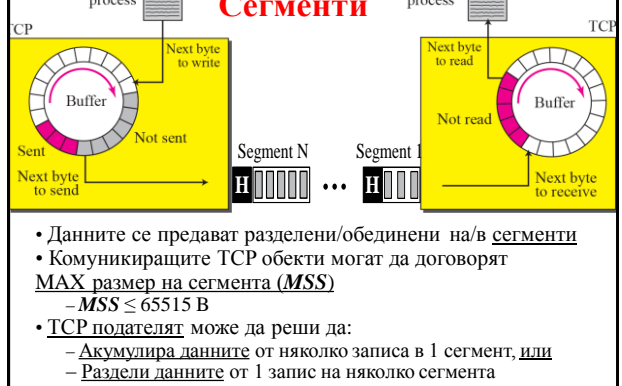
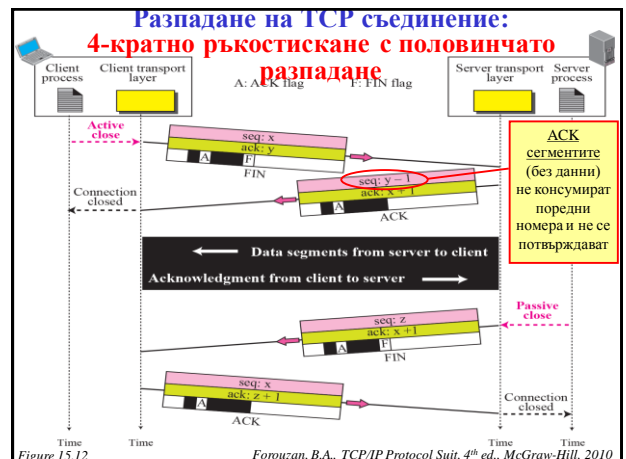
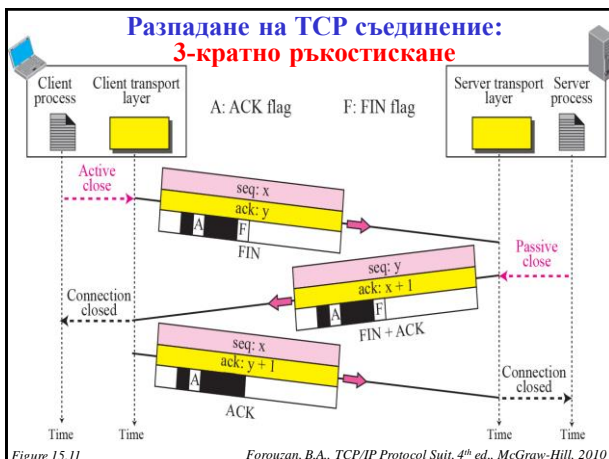
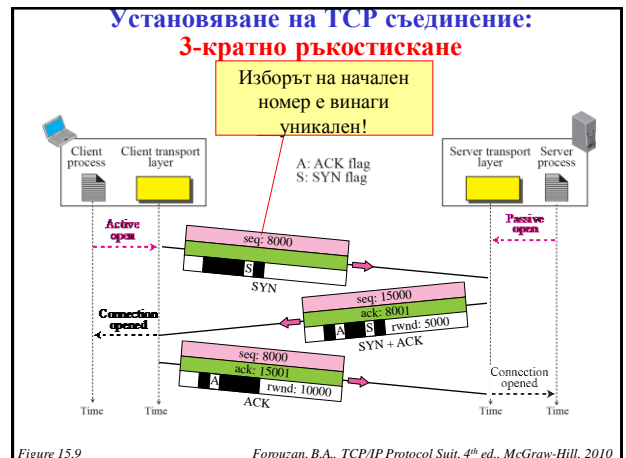
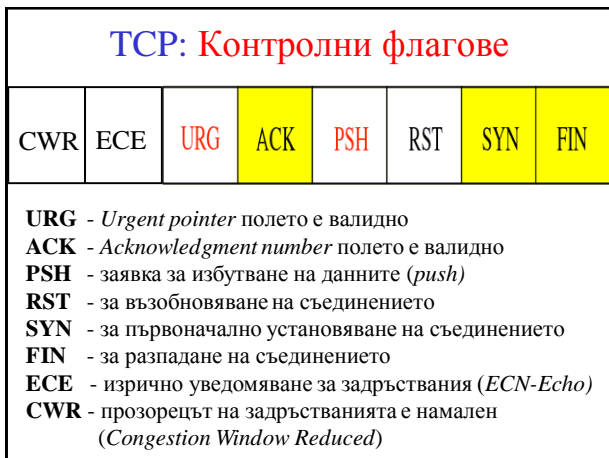
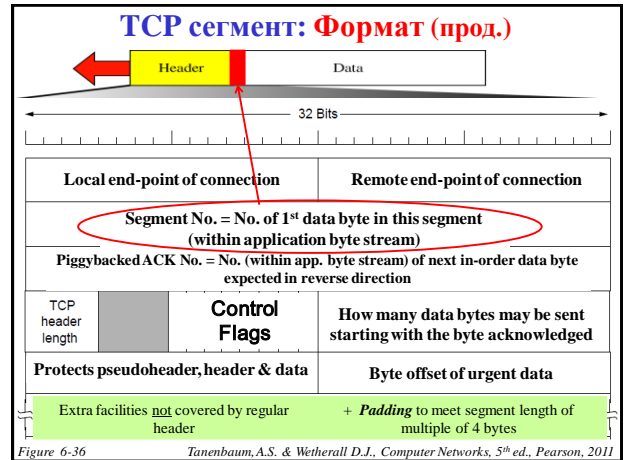
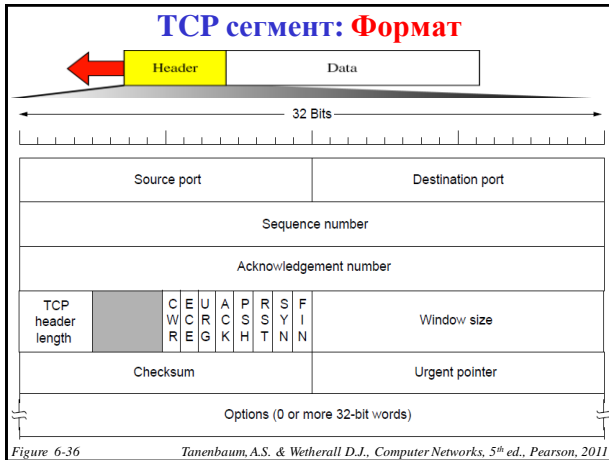


Figure 15.4

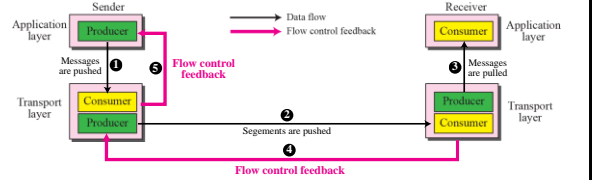
Forouzan, B.A., TCP/IP Protocol Suit, 4th ed., McGraw-Hill, 2010

- Данните се предават разделени/обединени на/в сегменти
- Комуникаращите TCP обекти могат да договорят MAX размер на сегмента (MSS)
 - $MSS \leq 65515$ B
- TCP подателят може да реши да:
 - Акумулира данните от няколко записа в 1 сегмент, или
 - Раздели данните от 1 запис на няколко сегмента



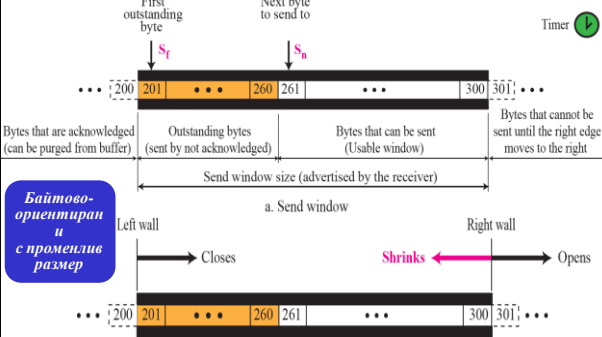
ТСР контрол на потока

ТСР контрол на потока: Обратна връзка



Forouzan, B.A., TCP/IP Protocol Suit, 4th ed., McGraw-Hill, 2010

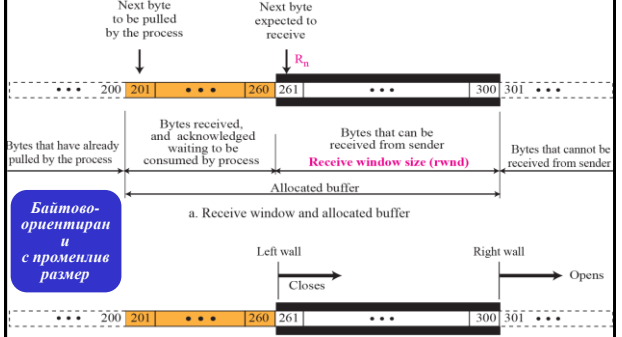
ТСР контрол на потока: Прозорец на подателя



- Подател b. Opening, closing, and shrinking send window
 - Не е длъжен да изпраща целия прозорец наведнъж
 - Не е длъжен да изпраща данните веднага след пристигането им от горния слой
- 15.22 Forouzan, B.A., *TCP/IP Protocol Suit*, 4th ed., McGraw-Hill, 2

Figure 15.22 горняя слой Forouzan, B.A., TCP/IP Protocol Suit, 4th ed., McGraw-Hill, 2010

ТСР контрол на потока: Прозорец на получателя



- Получател b. Opening and closing of receive window
 - Не е длъжен да изпраща веднага потвърждение
 - Може да изпрати потвърждение по всяко време

Figure 15.23 Forouzan, B.A., *TCP/IP Protocol Suit*, 4th ed., McGraw-Hill, 2010

ТСР контрол на потока: Пример

Note: We assume only unidirectional communication from client to server. Therefore, only one window at each side is shown.

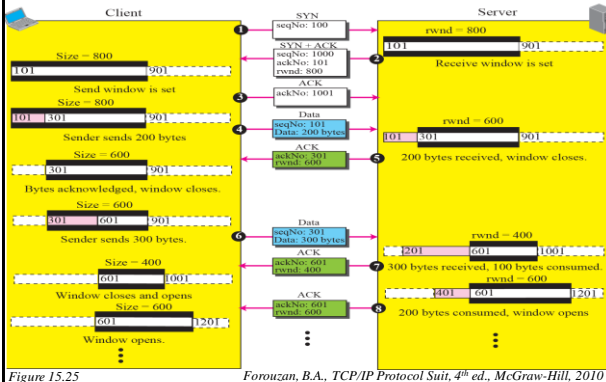


Figure 15.25 Forouzan, B.A., TCP/IP Protocol Suit, 4th ed., McGraw-Hill, 2010

ТСР: Управление на прозореца

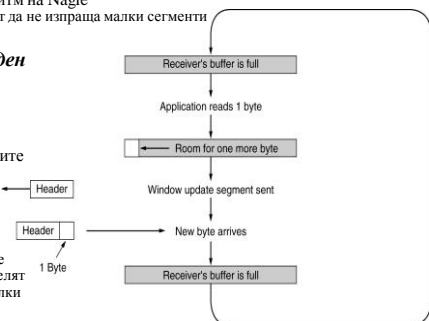
- Не е пряко обвързано с АСК механизъм
 - Специфична особеност на протоколите от транспортния слой
 - Размерът на прозореца може да се увеличава или намалява от **получателя** динамично
 - Въз основа на размера на буфера му
 - Ако буферът му е пълен, получателят анонсира нулев размер на прозореца.
 - Подателят спира да предава сегменти
 - 2 изключения
 - Спецични данни (*urgent*) могат да се предават (например, за прекратяване на отдалечен процес)
 - Специален сегмент-проба се изпраща периодично за да се предупреди получателя, че подателят все още очаква анонсиране на ненулев размер на прозореца.
-
- The diagram illustrates TCP flow control. It shows a Sender and a Receiver connected by a network. The Sender sends Segment 1 (seq: 1001, 4000 bytes). The Receiver's buffer is shown as a yellow bar with a total capacity of 4000. The buffer is currently full (4000 bytes used). The Receiver sends back 'ack: 5001' and 'win: 0', indicating that the buffer is full and no more data can be received. The Sender then sends Segment 2 (seq: 5001, 1000 bytes). The Receiver's buffer is shown again, now with 1000 bytes used and 3000 bytes free. The Receiver sends back 'ack: 5001' and 'win: 1000', indicating that the buffer has space for 1000 more bytes. The diagram shows the flow of data and the state of the receiver's buffer over time.

TCP: Синдром на глупавия прозорец

- Възниква, когато данните се обработват бавно от приложенията.
- **Синдром, създаден от подателя.**
 - Когато предаващото приложение генерира данните бавно
 - Т.е. само по няколко байта наведнъж.
 - Решение: Алгоритм на Nagle
 - Цел: подателят да не изпраща малки сегменти

- **Синдром, създаден от получателя.**

- Когато получаващото приложение консумира данните бавно
- 2 решения:
 - Алгоритм на Clark
 - Забавено потвърждение
 - Цел: получателят да не иска малки сегменти


**Синдром, създаден от подателя:
Алгоритм на Nagle
(прилаган от подателя)**

- Стъпка 1
 - Изпращане на 1. порция данни, получени от предаващото приложение.
 - Дори, ако това е само 1 байт.
- Стъпка 2
 - Акумулиране на данни в изходящия буфер докато:
 - Получателят изпрати обратно потвърждение (ACK), или
 - Се съберат достатъчно данни за запълване на:
 - Сегмент с максимален размер (MSS), или
 - ½ от прозореца на подателя
 - Изпращане на сегмент
- Стъпка 3
 - Повторяне на стъпка 2 до края на комуникацията

**Синдром, създаден от получателя:
Алгоритм на Clark
(прилаган от получателя)**

- Изпращане на потвърждение, веднага след пристигане на сегмент.
 - За да се подпомогне точното изчисление/актуализиране на RTT от страна на подателя
- Но се анонсира нулев размер на прозореца, докато във входящия буфер не се освободи достатъчно място, равно на:
 - 1 MSS, или
 - ½ от буфера

$$\text{available buffer space} \geq \min\left(\frac{\text{buffer size}}{2}, \text{maximum segment size}\right)$$

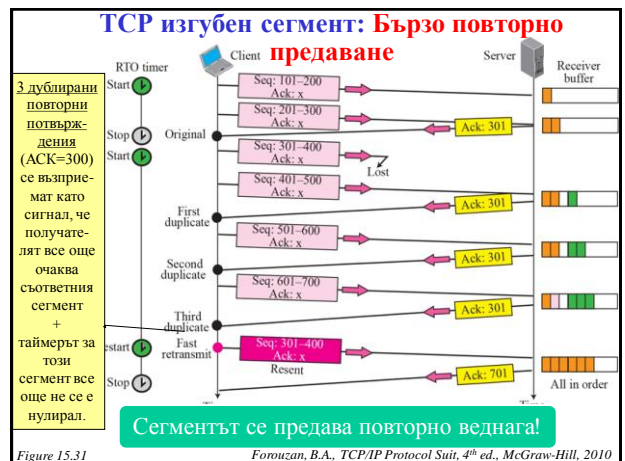
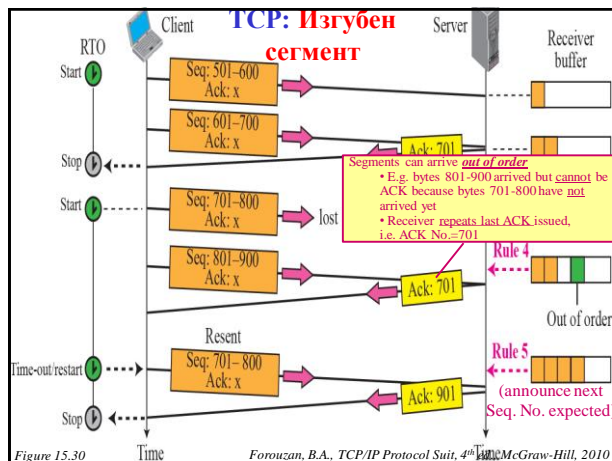
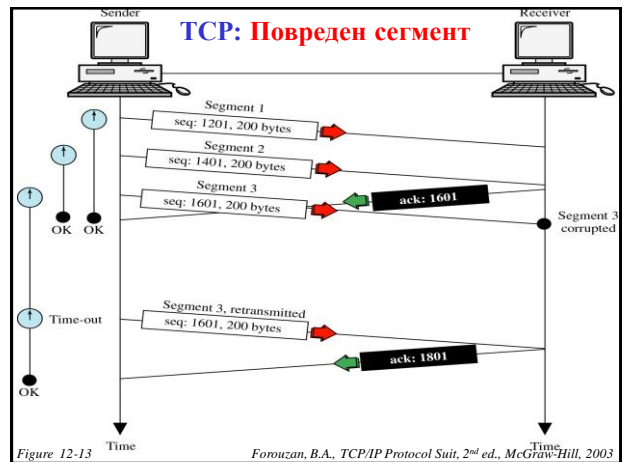
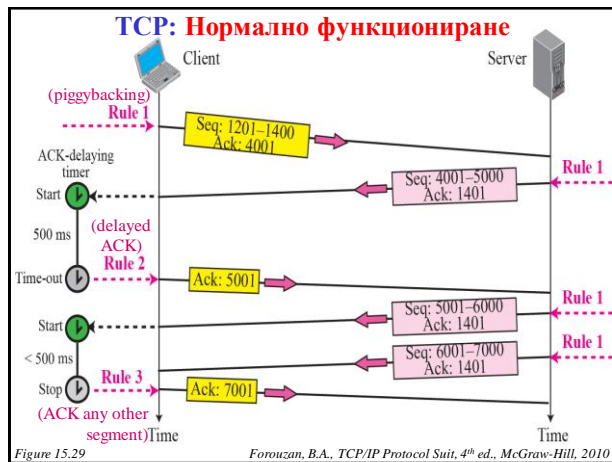
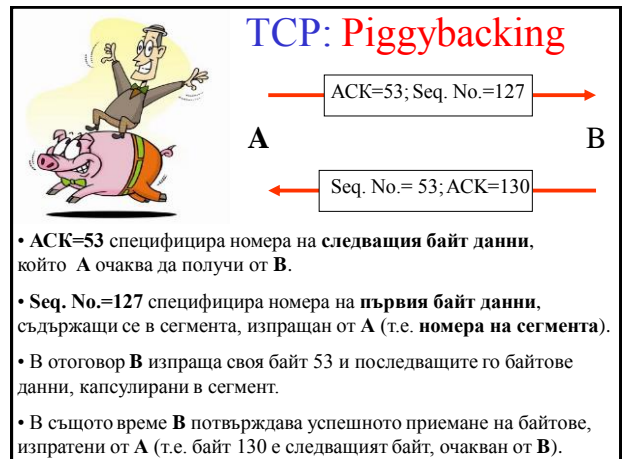
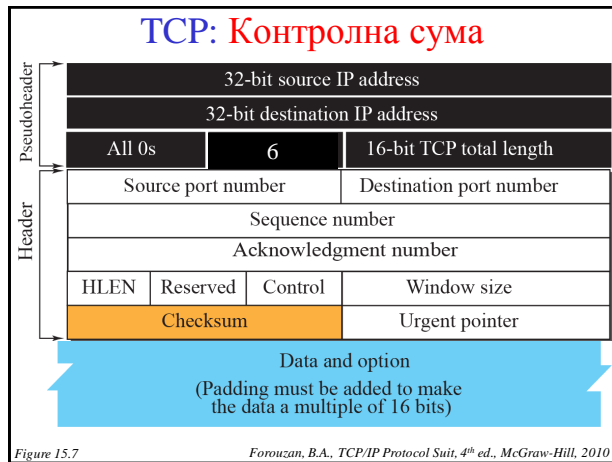
**Синдром, създаден от получателя:
Забавено потвърждение
(прилагано от получателя)**

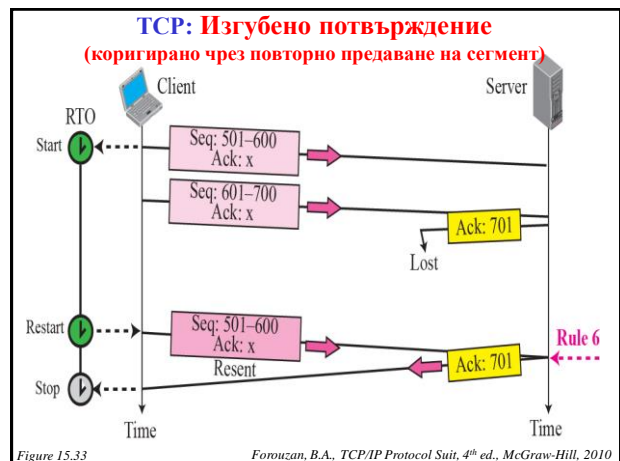
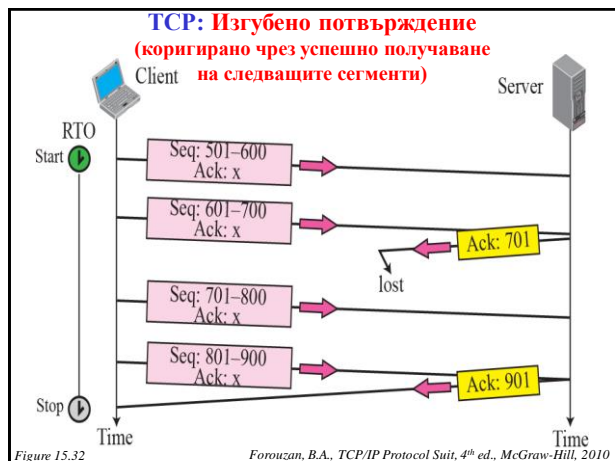
- Забавя се изпращането на потвърждение
- Докато не се освободи достатъчно място във входящия буфер
- **Предимство**
 - Намален трафик
 - Получателят не е длъжен да потвърждава всеки сегмент (особено, ако няма данни за предаване в обратната посока)
- **Недостатък**
 - Забавено потвърждение може да принуди подателя да изпрати отново някой сегмент
 - Затова потвържденията се бавят не повече от 500 ms

TCP контрол на грешките

TCP: Контрол на грешките

- Базиран на **PAR** схема
 - Положителна квитанция с повторно предаване
 - Positive Acknowledgement with Retransmission
- Всеки TCP сегмент съдържа **контролна сума** в заглавната си част
 - За защита от грешки на целия TCP сегмент (вкл. псевдозаглавната част)
- Получател
 - Ако сегментът не е повреден, го потвърждава с ACK.
 - Кумулативни потвърждения
 - Piggybacking
 - Сегментите, използвани само за потвърждение (т.е. без данни), не консумират поредни номера и не се потвърждават.
 - Ако сегментът е повреден, го отхвърля без изпращане на отрицателна квитанция (NAK).
 - Ако сегментът пристигне не поред, отлага ACK, докато не пристигнат всички липсващи сегменти.
- Подател
 - Ако таймерът се нулира преди пристигането на ACK, предава повторно сегмента.
 - Тъй като или сегментът е повреден или изгубен, или потвърждението му се изгубило.





TCP контрол на грешките: Проблеми

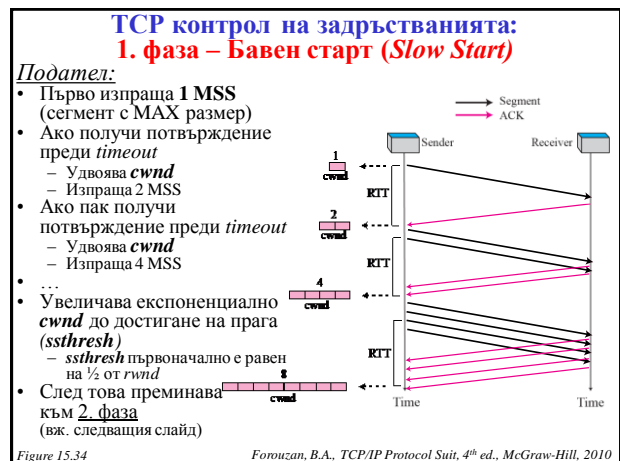
- Сегментите могат да се забавят при транзита
 - Може да се наложи повторното им предаване
 - Но с различен диапазон от байтове в сравнение с първоначалното им предаване
- Изисква се внимателно администриране
 - За да се среди кои байтове са били правилно доставени досега

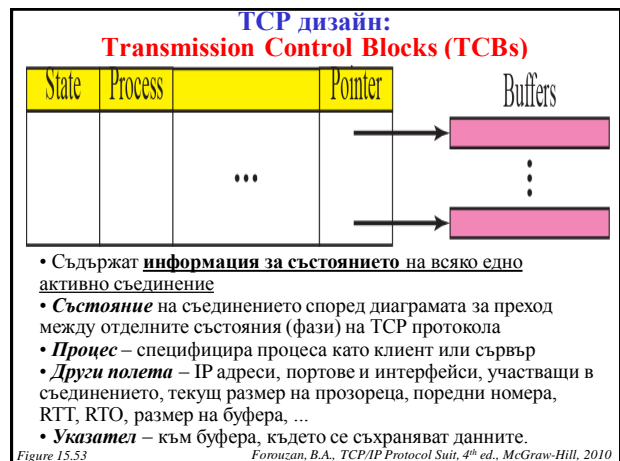
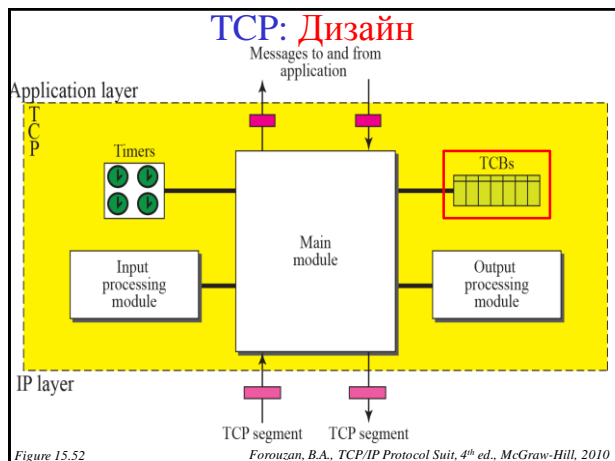
TCP контрол на задръстванията



TCP: Контрол на задръстванията (прод.)

- Основно допускане от страна на TCP
 - Изтичане на времето за изчакване (*timeout*) се причинява от задръствания, а не поради грешки в IP пакетите.
- Вярно в кабелните мрежи
 - Комуникационните линии в днешно време са много надеждни (напр. при използване на влакнесто-оптични кабели)
 - Много рядко пакетите пристигат с грешки
- Погрешно в безжичните мрежи
 - Изключително ненадеждни
 - Пакети се губят или пристигат с грешки винаги, по всяко време.
 - Друг подход е необходим за контрол на задръстванията



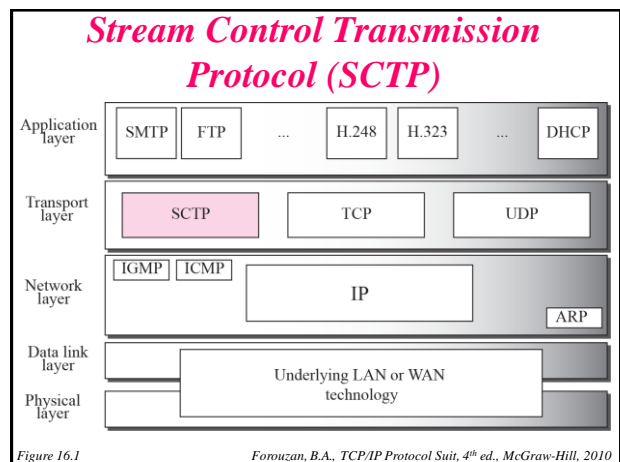


TCP: Реализации

Measure	RFC 1122	TCP Tahoe	TCP Reno	NewReno
RTT Variance Estimation	✓	✓	✓	✓
Exponential RTO Backoff	✓	✓	✓	✓
Karn's Algorithm	✓	✓	✓	✓
Slow Start	✓	✓	✓	✓
Dynamic Window Sizing on Congestion	✓	✓	✓	✓
Fast Retransmit		✓	✓	✓
Fast Recovery			✓	✓
Modified Fast Recovery				✓

- Бързо възстановяване (*fast recovery*)
 - В случай на бързо повторно предаване, прозорецът се намалява наполовина и директно се преминава към 2. фаза (плавна нарастване на прозореца)
- Модифицирано бързо възстановяване
 - Подобрена реакция при загуба на 2 сегмента от един прозорец

Table 22.5
Stallings, W., Data and Computer Communications, 9th ed., Prentice Hall, 2011



SCTP

- Дефиниран в RFC 2960
- Първоначално стандартизиран от IETF SIGTRAN WG за транспортиране на SS7 сигнализация през IP мрежи
- По-късно еволюира до транспортен протокол с общо предназначение, осигуряващ надеждно, пълно-дуплексно съединение с подобрени опции за доставка.
 - Необходим за нови приложения като IP телефония (VoIP), ISDN over IP, media gateway control, ...

Protocol	Port Number	Description
IUA	9990	ISDN over IP
M2UA	2904	SS7 telephony signaling
M3UA	2905	SS7 telephony signaling
H.248	2945	Media gateway control
H.323	1718, 1719, 1720, 11720	IP telephony
SIP	5060	IP telephony

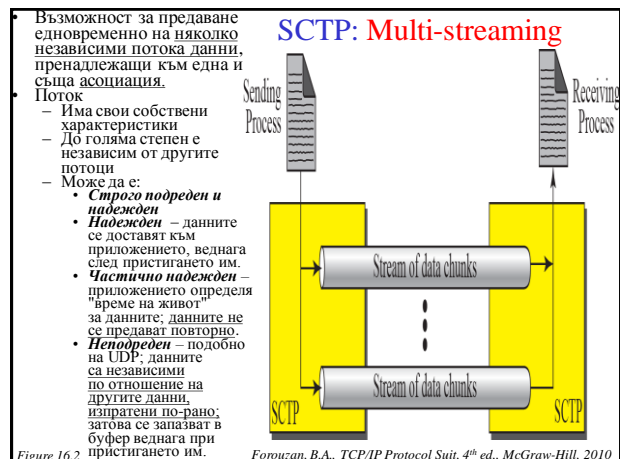
Table 16.1
Forouzan, B.A., TCP/IP Protocol Suit, 4th ed., McGraw-Hill, 2010

SCTP: Прилики с TCP

- **Логическо съединение**
 - Асоциация
- **Пълен дуплекс**
- **Надежден транспорт**
- **Поредни номера** за контрол на потока и на грешките
 - Същия диапазон като TCP, т.е. $(0, 2^{32}-1)$.
- **Piggybacking**
- **Контролна сума**
 - 32 бита (SCTP) / 16 бита (TCP)
- **Бързо повторно предаване** в случай на:
 - 4 SACKs (SCTP) / 3 ACKs (TCP)
- **Бързо възстановяване**
- **Контрол на задръстванията** (същите фази като TCP)
 - Но прилагани поотделно за всеки поток (*stream*)

SCTP: Разлики с TCP

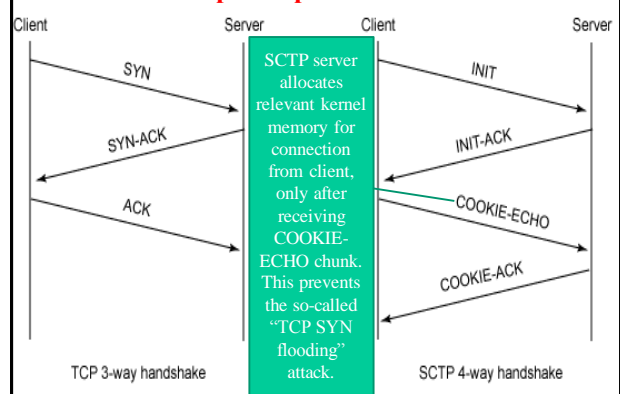
- **Ориентиран към съобщения, а не към байтове.** (запазва границите на съобщенията на приложния слой)
- **Номерираща парчета от данни (chunks)**
 - TCP номерира байтове
- **Последователни номера се използват само за потвърждение на данни (data chunks)**
 - Control chunks се потвърждават с други control chunks
- **Потвърждава последния получен номер, не следващия очакван номер.**
 - Т.е. ACK_x означава 'Изпратих X+1'
- **Няма опции в заглавната част**
 - Опциите се задават чрез дефиниране на нови chunk видове
- Поддържа на **множество потоци от данни (multi-streaming)** – ако един поток е блокиран, останалите продължават да доставят данни, аналогично на магистралните платна.
 - TCP използва само 1 поток от данни
- **Multi-homing** (хостовете могат да дефинират няколко IP адреса / мрежови интерфейси във всеки край на комуникацията за отказоустойчивост, ако един от пътищата за доставка се провали, започва използването на друг)
 - TCP използва само по 1 IP адрес във всеки край на комуникацията



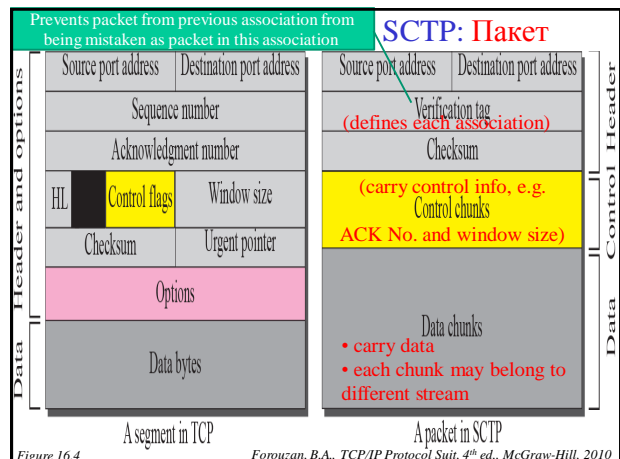
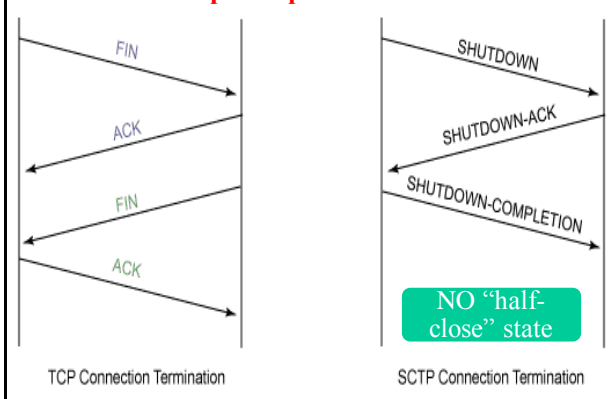
SCTP vs. TCP & UDP

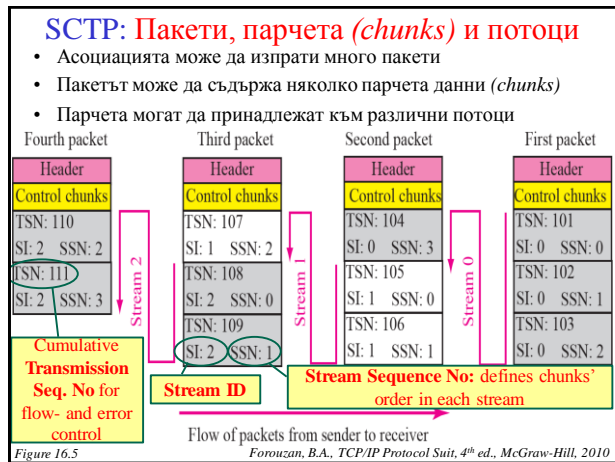
Feature	UDP	TCP	SCTP
Connection oriented	No	Yes	Yes
Reliable transport	No	Yes	Yes
Unreliable transport	Yes	No	Yes
Preserve message boundaries	Yes	No	Yes
Ordered delivery	No	Yes	Yes
Unordered delivery	Yes	No	Yes
Data checksum	Yes	Yes	Yes
Checksum size (bits)	16	16	32
Path MTU	No	Yes	Yes
Flow-, error-, congestion control	No	Yes	Yes
Multiple streams	No	No	Yes
Multi-homing support	No	No	Yes

SCTP установяване на съединение: 4-кратно ръкостикане



SCTP разпадане на съединение: 3-кратно ръкостикане





SCTP: Multi-homing

- Хостове
 - Могат да използват няколко IP адреса
 - Обвързани едновременно с една асоциация
- SCTP асоциация
 - Може да използва множество налични IP адреси
 - Setup (*INIT* chunk) – всяка крайна точка посочва списък с налични IP адреси
 - Setup (*INIT-ACK* chunk) – крайните точки избират двойка валидни IP адреси като основни (*primary*), между които се осъществява предаването на данните по подразбиране (всички други адреси се маркират като вторични / *secondary*)
- Път за предаване
 - Наблюдаван чрез изпращане на HEARTBEAT chunks, за изпробване достижимостта на даден адрес.
 - Счита се за неактивен, ако предаването по него се провали многократно (данните се предават повторно към избран вторичен адрес).
 - Приложението се информира за състоянието на пътя (в случай на промени)
 - Приложението може да инструктира SCTP да използва друг път за основен

