Тема 16. Протоколи от транспортния слой – TCP и UDP

TCP (Transmission Control Protocol)
  Откриване на TCP сесия
  Закриване на TCP сесия
  Портове и сесии
UDP (User Datagram Protocol)

Двата протокола TCP (Transmission Control Protocol) и UDP (User Datagram Protocol) работят в транспортния слой на модела DoD. Те са отговорни за транспортирането на данните в мрежата. При предаване на съобщение по мрежата може да се използва единият или другият протокол. Когато е необходимо да се гарантира получаването на данните по мрежата се използва TCP. Когато скоростта на трансфер е с по-висок приоритет се използва UDP.

Настоящата тема разглежда основните характеристики, портовете на двата протокола, случаите, при които се предпочита негарантирана пред гарантирана доставка, приложенията които използват единия или другия протокол.

TCP (Transmission Control Protocol)

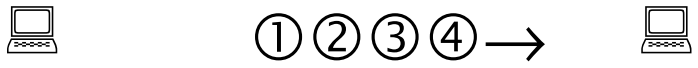
Протоколът TCP осигурява зависим от връзката трафик между две устройства в мрежата. Когато се изпраща съобщение, между компютъра, който изпраща и компютъра получател трябва да се изгради сесия (връзка). Съобщението, което се изпраща се разделя на сегменти, като всеки сегмент има пореден номер. При достигането на сегментите при получателя, те се подреждат по тези номера, за да се получи оригиналното съобщение. Ако някой от сегментите не е пристигнал се изисква неговото повторно изпращане. Максималният брой препредавания на сегмент се определят от параметъра TcpMaxDataRetransmissions (стойност по подразбиране 5 пъти). Виж: Структура на TCP сегмент.

Протоколът TCP осигурява гарантирано достигане на информацията до получателя. За да направи това, освен необходимите за изпращане данни се включва допълнителна служебна ‘статус’ информация. Този статус включва съобщения за потвърждаване на получаването (ACK), рестартиране – изчистване на буферите (RST), синхронизация на последователността (SYN), край на сесия (FIN) и др. Потвърждаването става чрез изпращане на сегмент статус ACK съдържащ следващия поредния номер.

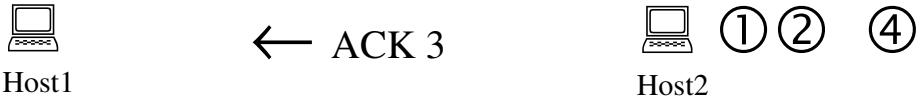
Потокът информация получена от приложението се разделя на сегменти в зависимост от параметъра MSS (Maximum Segment Size). Сегментите се поставят в буфер – наречен прозорец (с размер – параметър TCPWindowSize). Изпращачът изпраща всички сегменти съхраняващи се в прозореца. За всеки сегмент се съхранява брояч RTO (retransmission timeout). Изпращачът препредава сегмент, ако не получи потвърждение за RTO секунди. Това време се определя динамично и зависи от качеството на връзката. При изграждането на сесията стойността RTO е 3 секунди. При успешно предаване на сегменти последователно, RTO се намалява. При всяко препредаване на сегмент RTO се удвоява.

Пример:

1. Host1 изпраща информация на Host2 – 4 броя сегменти, номерирани 1, 2, 3 и 4.



2. Host2 получава сегменти 1, 2 и 4. Изпраща сегменти 1 и 2 – успешно получени – на програмата от по-високо ниво. Изпраща потвърждение ACK с номер на сегмент 3.



3. Host1 изпраща сегменти 3 и 4

4. Host2 получава сегменти 3 и 4. Сегмент 4 е повторно изпратен и съответно игнориран. Съобщение за грешка не се генерира. Към програмата от високо ниво се предава информацията от сегментите 3 и 4. Изпраща се потвърждение ACK с номер на сегмент 5.

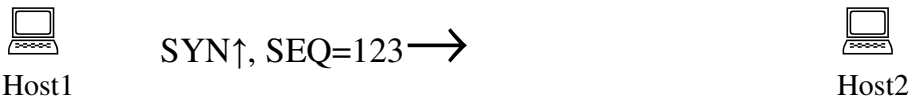
Виж: Таймер за забавено потвърждение

Откриване на TCP сесия

Сесията е виртуален канал, през който две приложения комуникират. Когато едно устройство инициира сесия, то предава сегмент с флаг SYN=1 (синхронизиране, SYN↑) и изпраща текущия пореден номер сегмент. Ако насрещното устройство се съгласи да комуникира, то изпраща сегмент с флаг ACK със следващия пореден номер. За установяване на двупосочна комуникация (пълен дуплекс) при този механизъм би трябвало да се използват общо четири сегмента. В TCP това става с три сегмента – т.нар. тристранно TCP договаряне. Икономията на един сегмент идва от обединяване в един сегмент на съобщението от насрещното устройство за потвърждаване и синхронизация.

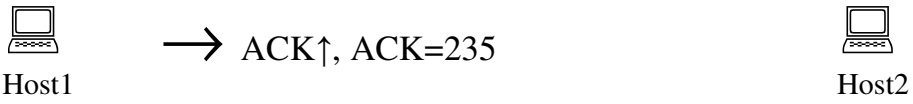
Пример:

1. Host1 изпраща сегмент с флаг SYN и пореден номер SEQ (Sequence Number=123).



2. Host2 отговаря с флаг ACK и следващия пореден номер, който очаква (Acknowledgement Number=124). В същия сегмент се вдига флагът SYN и се попълва поредния номер за Host2 (Sequence Number=234).

3. Host1 потвърждава с флаг ACK (Acknowledgement Number=235).

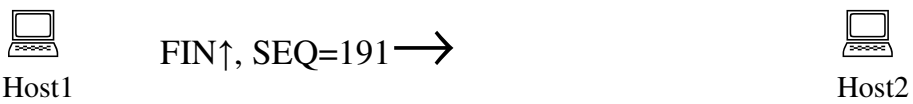


Закриване на TCP сесия

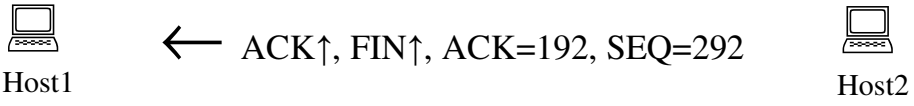
Сесията нормално се закрива и от двете крайни точки. Когато приложението затваря комуникационната сесия TCP изпраща сегмент с флаг FIN=1 (FIN↑).

Пример:

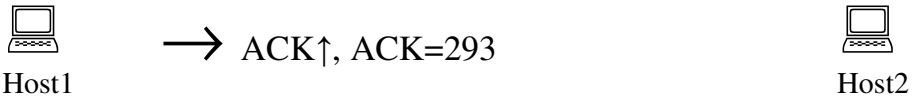
1. Host1 изпраща сегмент с флаг FIN и пореден номер SEQ (Sequence Number=191).



2. Host2 отговаря с флаг ACK и следващия пореден номер, който очаква (Acknowledgement Number=192). В същия сегмент се вдига флагът FIN и се попълва поредния номер за Host2 (Sequence Number=291).



3. Host1 потвърждава с флаг ACK (Acknowledgement Number=293).



Виж: Статус на TCP сесия

Портове и сесии

Протоколите TCP и UDP са посредници между приложенията и IP протокола. Съобщенията се изпращат на устройство, по зададен IP адрес. На съответното устройство (или компютър) работят много програми. Необходимо е да се укаже коя програма да бъде получател на съобщението. Това става чрез задаване на номер на ***порт.***

Портът представлява число от 0 до 65 535. За да се осъществи комуникация, в устройството получател е стартирана програма, която „слуша" определен порт. Портовете с малки номера (от 0 до 1 023) са наречени добре познати портове (well-known ports), присвоени на конкретни приложения от организацията IANA. Например, портът за HTTP е 80, за FTP е 21 и т.н. Всяка операционна система съдържа текстов файл с познатите портове. В Windows XP и следващи версии този файл се намира в директория '\windows\system32\drivers\etc' и се нарича 'services'.

За определяне на крайната точка на TCP сесия се използва ***сесиен адрес***. Той се състои от IP адреса и номера на порта. При изписване номера на порта се разделя от IP адреса със символ двоеточие. Примерно: 194.145.63.12:80 е сесиен адрес на WEB услугата на 'www.dir.bg'.

При установяване на TCP сесия от сървър се извършва пасивно отваряне (passive open). Така приложението-сървър указва на операционната система кой порт желае да приема връзки. Портът се намира в състояние на приемане (listening state).

Програма в устройство-клиент изисква от операционната система да се свърже към сесиен адрес. Това е активно отваряне (active open), открива се сесията. Програмата клиент също трябва да притежава сесиен адрес. Операционната система служебно задава номер на порт, случайно число, по-голямо от 1024.

Състоянието на изградените сесии може да се наблюдава чрез TCP помощната програма netstat.

UDP (User Datagram Protocol)

Протоколът UDP осигурява негарантирана доставка на данни. UDP осъществява изпращането на данните и не се занимава с проверка на получаването им. Съобщението, което се предава се нарича дейтаграма (datagram). Теоретически, максималната дължина на UDP дейтаграмата е 65515 байта. Такава дейтаграма задължително ще бъде фрагментирана на по-малки пакети или отхвърлена ако фрагментирането е забранено. Използването на големи дейтаграми в глобалните мрежи не е желателно, тъй като е възможно загубването или дублирането на фрагменти. UDP осигурява единствено контролна сума, гарантираща целостта на данните.

Програмите, използващи UDP протокола трябва да реализират самостоятелно:

- препредаване на загубени дейтаграми;
- игнориране на повторения;
- фрагментиране и обединяване на големи потоци данни.

В локалните мрежи грешките при предаване на информация са пренебрежимо малки. Използването на UDP протокол ще генерира по-малък трафик. Обикновено, при избора на UDP протокол се търси не намаляване на трафика, а по-скоро намаляване на времето за доставка и отговор. Например, при използване на UDP при предаване на глас, загубата на сегмент ще доведе до дефект – пукане. Използването на TCP при същото приложение ще доведе до забавяне – накъсване на звука, сериозно забавяне времето за отговор.

UDP е транспортен протокол за DNS, SNMP и TFTP. UDP се използва при пренасяне на глас по Интернет (VoIP), интернет радио, реално видео и аудио. Услугите multicast и broadcast могат да се реализират единствено чрез UDP.

Виж: Структура на UDP сегмент, anycast, broadcast, multicast, unicast

Въпроси:

1. Коя е основната разлика между протоколите TCP и UDP?
2. Как работи протоколът TCP?
3. Колко сегмента са необходими при откриване на една TCP сесия?
4. Какво е предназначението на порта?
5. При кои случаи е удачно да се използва UDP протоколът вместо TCP?