

## Проект по Компютърни мрежи

### Изготвил: Биляна Инджева 4MI0800106

#### Задание:

Искаме да изтеглим файл посредством “торенти”. Опишете цялостния процес по осъществяването на тази комуникация. Приемаме следната постановка:

- Използва се току-що конфигурирана компютърна мрежа свързана към интернет.(Всички свързани устройства са активни и функционални).
- Големината на файла , който трябва да се свали е 512 MB. Да се опише какво трябва да е известно преди да може да се осъществи комуникация, да се опише какво е необходимо да се намери преди да се направи опит за комуникация, какво се случва по време на комуникацията, като се споменат на кратко използваните протоколи по съответстващите слоеве на OSI модела и основните им особености (включително портове). По желание може да се разгледа същият случай с добавяне на: NAT, VPN, цифрови сертификати, Firewall, DMZ зони или разпадане на връзката по време на комуникацията.

Торент системата е метод за сваляне на файлове, който използва BitTorrent протокола. Вместо да се ползва централен сървър, информацията се изтегля от децентрализирана мрежа на други потребители, наречени сийдъри. Сийдърите споделят части от желания файл, които лийчърите (потребителите, които свалят файловете) изтеглят.

Когато лийчърът изтегли всички части, те се събират и се сглобява файлът, позволявайки на потребителя да го използва и споделя с други участници в мрежата. За да може файлът да бъде изтеглен, трябва да се намира наличен в системата, тоест да се намира на поне едно от устройствата на сийдърите, които са съгласни да го споделят с другите.

Когато сийдърът реши да сподели определен файл в системата за изтегляне, той трябва да създаде и изпрати до торентния сървър един торент файл. Този торент файл съдържа информация за файловете, които се споделят, както и списък с адресите на сийдърите в мрежата. Лийчърите използват торентния файл, за да се свържат със сийдърите и да започнат свалянето на файловете.

Така торент системата и BitTorrent протоколът позволяват на потребителите да свалят файлове от други участници в мрежата, като разпределят натоварването и ускоряват процеса на сваляне.

След достигането на торента до сървъра, първият сийдър може да бъде открит чрез тракера, а от него може да се свали файлът. Това увеличава наличността на файла в мрежата и подобрява неговото стабилно присъствие. Сега, след като разгледаме процеса на добавяне на нови файлове, нека се фокусираме върху изтеглянето им. За начало, потребителят прави запитване към торентния сървър посредством търсачка (search engine), за да провери наличието на желания файл в мрежата. Ако файлът е наличен, потребителят го изтегля от сървъра чрез HTTP/HTTPS. Това му осигурява съответния торент файл, отговарящ на неговите изисквания. С използването на този торент файл, потребителят може да се свърже с тракера и да получи информация за активните сийдъри. След като определени са сийдърите, потребителят се свързва с тях и започва да изтегля различните части на файла, използвайки TCP протокола. За да се гарантира цялостта на данните, към всеки фрагмент се добавя проверка за грешки (checksum), след което данните се инкапсулират и се включва контролна информация в хедъра. Така всяка част се превръща в пакет от данни (Protocol Data Unit - PDU), който се предава успешно посредством Internet Protocol. Пакетите се препращат чрез процес на маршрутизация, като се избира оптималният маршрут, който да следват. В този случай, link-state базиран протокол е по-подходящ от distance vector, тъй като при link-state се създава пълна маршрутна карта на едно място, а потребителите се осведомяват за своето съществуване чрез изпращане на по-малки пакети, които са несравними по размер с маршрутните таблици. Този механизъм работи добре в голямата среда с множество потребители. След като всички части достигнат до лийчъра, те се разопаковат и се подредят така, че файлът да бъде възстановен в началното си състояние. Важно е да се отбележи, че по време на или след изтеглянето потребителите могат да станат и сийдъри.

След анализа на процеса се появяват няколко интересни въпроса, първият от които е как се определят парчетата, които трябва да бъдат изтеглени първо. В общи линии, идеята е да се предпочитат първо най-рядко срещаните парчета, за да се гарантира, че няма да се загуби някое от тях. За да дадем по-подробен отговор, трябва да разгледаме някои особености на протоколите BitTorrent и TCP.

Важно е да отбележим, че всяко парче от файл се разделя на по-малки подпарчета с размер обикновено от 16KB до 1MB. В даден момент има винаги налични заявки за нови подпарчета (най-често около 5), като след като е изпълнена заявка, веднага се добавя нова. Трябва също да се отбележи механизмът на бавен старт (slow start) в TCP, който регулира скоростта на мрежата, като постепенно увеличава пропускателната способност до достигане на максимума.

Със знанието за тези факти, можем да изброим основните правила на алгоритъма за избиране на парчета:

- При тегленето на подпарчето се изисква изтеглянето на всички останали подпарчета от същото парче преди да се започне изтеглянето на нови такива.
- Винаги се тегли най-рядкото парче, което допринася за разрастването на наличността на файла. Това увеличава скоростта на теглене и нашата капацитет за теглене, тъй като рядките парчета имат повече желаещи да ги изтеглят от нас. В резултат, ние също можем да теглим повече.
- Първото парче винаги се избира произволно, с цел да бъде често срещано. Различните парчета имат различни скорости на теглене, и в началото на връзката TCP протоколът не е достигнал оптималната си скорост. Е важно потребителят да започне да притежава парчета възможно най-рано, за да може да се включи в мрежата и да играе ролята на сийдър.
- Когато на конвейера със заявки остане само едно парче за изтегляне, клиентът изпраща заявки до всички пиъри в мрежата. След като се открие, кой притежава парчето и клиентът го получи, останалите заявки се отказват. Това се прави, за да се избегне ситуацията, в която последното парче да бъде изтеглено от потребител със забавена скорост на трансфер.

Избирането от кои сийдъри да се изтегля в даден момент се случва чрез "tit-for-tat" алгоритъмът, който се основава на следните правила и разглежда комуникацията между двама потребители като ходове:

- При установяване на контакт с друг потребител в началото, винаги се кооперираш (т.е. предлагаш сътрудничество и подкрепа).
- На всеки следващ ход се държиш спрямо другия потребител така, както той се е държал спрямо теб в предишния ход. Ако той ти предостави подпомагане или тегли от теб, ти правиш същото към него.
- Когато отсрещната страна извърши нещо несправедливо или наруши правилата спрямо теб, след адекватен отговор, трябва да бъдеш готов да простиш и да продължиш да се държиш справедливо към нея.

Този алгоритъм на "tit-for-tat" има за цел да поддържа балансирана и справедлива връзка между сийдърите, като се стимулира сътрудничеството и споделянето на информация в мрежата.

В BitTorrent системата за определяне на потребителите, с които активно ще си кооперираме или за реакция на несправедливо държание, можем да използваме Choking Algorithm (алгоритъм за блокиране), който ни позволява

временно да откажем да предоставяме нашите парчета на даден потребител, без да прекъсваме тегленето от него. Това действие се нарича "choking", докато за потребителите, с които решим да си сътрудничим, казваме, че са били "unchoke" (разблокирани).

В даден момент има фиксиран брой на "unchoke" потребителите (по подразбиране 4), като кои да бъдат те се определя от скоростта на теглене от тях. За да предотвратим честотното "choking" и "unchoking", което може да се стигне при бавния старт на TCP протокола, използваме средната скорост на трансфер през последните 20 секунди.

Това означава, че като лийчър (потребител, който само тегли), за да ни бъде позволено да теглим от други потребители (да бъдем "unchoke"нати), трябва да генерираме достатъчен трафик на теглене от наша страна. Това означава, че трябва да бъдем активни сийдъри (потребители, които споделят информация). Това е един от най-важните аспекти на BitTorrent системата, тъй като стимулира потребителите да споделят и поддържа широко разпространение на файловете, като същевременно предотвратява натрупването на потребители, които само теглят, без да споделят.

Освен основните 4 "unchoke" потребителя, има още един, за когото се прилага "Optimistic Unchoking" (оптимистично разблокиране). Това означава, че периодично, на случаен принцип, избираме един потребител от мрежата и го "unchoke"ваме, без да се базираме на скоростта му на теглене. Това се прави с цел да дадем възможност на нови или по-бавни потребители да се утвърдят и да се развият като сийдъри.

Choking Algorithm и Optimistic Unchoking са важни механизми в BitTorrent системата, които подкрепят сътрудничеството, споделянето и балансирането на трафика между потребителите.

"Snubbing" е механизъм, който се използва в BitTorrent протокола, за да се справи със ситуацията, при която всички сийдъри, от които теглим, ни "choke"ват (блокират). Ако в продължение на последните 60 секунди не получим нищо от даден сийдър, той се счита за "snubbed" (пренебрегнат). Съгласно принципа на "tit-for-tat", от своя страна отказваме да му предоставяме информация (освен ако не е "optimistic choke"), след което увеличаваме броя на "optimistic choke" (оптимистични блокировки) за да намерим по-бързо нови сийдъри.

След приключването на изтеглянето, когато искаме само да споделяме (seed), използваме различен "choking" алгоритъм. Този алгоритъм "unchoke"ва потребителите с висока скорост на качване, т.е. добрите сийдъри, които често

споделят информация. Това гарантира, че файловете ще бъдат разпространени и репликирани по-бързо.

Интересното развитие на BitTorrent е появата на така наречените "trackerless" торенти. Вместо да използват частен или публичен тракер сървър, те търсят потребителите, които в момента теглят торента, чрез хеш таблица. Това позволява на потребителите да комуникират директно помежду си, без да се нуждаят от централизирана инфраструктура за проследяване на торента.