

ივანე ჯავახიშვილის სახელობის თბილისის სახელმწიფო უნივერსიტეტი

ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტი
კომპიუტერულ მეცნიერებათა დეპარტამენტი

ცოტნე ჩახვაძე

ერთრანგოვანი ქსელი (P2P), ბიტტორენტ (BitTorrent) კლიენტები

სამაგისტრო პროგრამა: ინფორმაციული სისტემები

სამაგისტრო ნაშრომი შესრულებულია ინფორმაციულ სისტემებში
მეცნიერების მაგისტრის აკადემიური ხარისხის მოსაპოვებლად

ხელმძღვანელები:

გია სირბილაძე,

ფიზ.-მათ. მეცნიერებათა დოქტორი,
სრული პროფესორი;

ირინა ხუციშვილი,

ფიზ.-მათ. მეცნიერებათა კანდიდატი,
ასოცირებული პროფესორი.

თბილისი

2019

ანოტაცია

BitTorrent არის (P2P) მონაცემთა გადაცემის პროტოკოლი. ის 2001 წლიდან არსებობს, მას შემდეგ რაც ბრენ კონმა შექმნა პროტოკოლის პირველი ვერსია. BitTorrent არის მონაცემთა ფართოდ გავრცელების მეთოდი, რომელშიც განსხვავებით კლიენტ-სერვერული ტრადიციული მეთოდისა არ გვხვდება ერთადერთი დისტრიბუტორი და მასში დატვირთვა გადანაწილებულია ყველა მონაწილე მხარეზე. ეს უკანასკნელი აღარ ემსახურება თითოეულ მიმღებს: მომხმარებელს უკვე მიღებული მონაცემებს უნაწილებენ სხვა მიმღებები. შესაბამისად, მონაცემთა გადანაწილების ღირებულება და დატვირთვა მნიშვნელოვნად მცირდება. ამავდროულად პროტოკოლი უზრუნველყოფს გადამეტებულ თავდაცვას აპარატურული პრობლემების თავიდან ასაცილებლად და ამცირებს მის დისტრიბუტორზე დამოკიდებულებას.

დიდი გარღვევა იყო, როდესაც პოპულარული გახდა ისეთი საიტები, როგორცაა: Pirate Bay და ა.შ. მოცემული საიტები საშუალებას გვაძლევს ჩამოვტვირთოთ პირატული მასალა. იმის გამო, რომ მომხმარებლები იყენებენ ტორენტ კლიენტებს ფილმების გადმოსაწერად, სტრიმინგ საიტები, როგორცაა Netflix ნელ-ნელა განიცდიან მომხმარებელთა შემცირებას. მიუხედავად ზემოთთქმულისა BitTorrent კვლავ გამოიყენება სხვადასხვა სამართლებრივი გადაწყვეტილებებში, სადაც დიდი ზომის ფაილების განაწილება მნიშვნელოვანია. სტატისტიკის მიხედვით, BitTorrent ინტერნეტ ტრაფიკის დაახლოებით 35% -ს შეადგენს. კანონირი მიზნებისთვის გამოყენებული გადაწყვეტილებები, შემდეგ ჩამონათვალში შეიძლება ვიხილოთ:

- Facebook იყენებს მას, რომ განაახლოს მათი დიდი მონაცემთა ცენტრები;
- Amazon S3 იყენებს მას, რათა ჩამოტვირთოს სტატიკური ფაილები;
- ტრადიციული ჩამოტვირთვები, რომლებიც უფრო გამოიყენება დიდი ფაილებისათვის, როგორცაა Linux დისტრიბუტივები და ა.შ.
- ზოგიერთი სათამაშო კომპანიები იყენებენ მას, როგორცაა: Blizzard, Eve და ა.შ.

BitTorrent ფართოდ გამოიყენება: განათლებაში, სახელმწიფო სექტორში და სხვ.

წარმოდგენილ სამაგისტრო ნაშრომში შექმნილია მულტიპლატფორმული, სრულყოფილი, BitTorrent პროტოკოლთან თავსებადი, კლიენტ პროგრამული უზრუნველყოფა, პროგრამირების ენა, პითონის ასინქრონული ბიბლიოთეკის გამოყენებით - asyncio. პროგრამა წარმოდგენილია როგორც გრაფიკული ინტერფეისით, ასევე ავტორის მიერ შექმნილი პროგრამის გამოყენება შესაძლებელია როგორც command line (cli) ბრძანებად.

Annotation

BitTorrent is a peer-to-peer (P2P) data transfer protocol. It exists since 2001 when Bram Cohen created the first version of the protocol. BitTorrent is a method of widely distributing large amounts of data by spreading the inherent load on hardware, Internet hosting and bandwidth when, in the traditional model, it is the sole distributor. The latter no longer has to serve each recipient: the customers themselves serve the data already received to new recipients. The cost and burden of data distribution are therefore considerably reduced; At the same time, the protocol provides redundancy against hardware problems and reduces dependence on the original distributor.

The big breakthrough was when sites like The Pirate Bay made it popular to use to download pirated material. Streaming sites, such as Netflix, may have resulted in a decrease in people using BitTorrent to download movies. But BitTorrent is still used in a number of different legal solutions where the distribution of large files is important. According to statistics, BitTorrent accounts for about 35% of Internet traffic. Its usage is shown below:

- Facebook uses it to distribute updates to their huge data centers
- Amazon S3 implements it for downloading static files
- Traditional downloads are still used for larger files such as Linux distributions
- Some Gaming companies use it, like Blizzard, Eve, etc.

It's widely used in Education, Government sector, Entertainment - Including pirating etc.

In this master's work author created TSO – multi-platform client software compatible with the BitTorrent protocol, using asynchronous Python module (**asyncio**). This program has graphical interface, but it also can used as command line program.

შინაარსი

1. შესავალი	5
2. ერთრანგოვანი ქსელი (P2P).....	6
2.1 ერთრანგოვანი ქსელის ტოპოლოგიები	7
2.1.1 წმინდა ერთრანგოვანი ქსელი	7
2.1.2 ჰიბრიდული ერთრანგოვანი ქსელი	7
3. BitTorrent - შესავალი.....	8
3.1 BitTorrent-ის ისტორია	9
3.2 გამოყენების სფეროები.....	10
4. BitTorrent არქიტექტურა	11
4.1 ალგორითმები	12
4.1.1 ნაწილის შერჩევის ალგორითმი	13
4.1.2 ქვენაწილები	13
4.1.3 Policy #1: Strict Policy	13
4.1.4 Policy #2: Rarest First	13
4.1.5 Policy #3: Random First Piece	13
4.1.6 Policy #4: Endgame mode	14
4.2 რესურსების გამოყოფა.....	14
4.2.1 Choking ალგორითმი	14
4.2.2 ოპტიმისტური „unchoking“	15
4.2.3 Anti-snubbing.....	15
4.2.4 მხოლოდ ატვირთვა.....	15
4.2.5 დეცენტრალიზებული ტრეკერი.....	15
5. ბიტტორენტის პერსპექტივები.....	16
6. პროგრამული უზრუნველყოფა.....	17
6.1 BitTorrent კლიენტის კონსოლური ინტერფეისის გამოყენების ბრძანებები:	19
6.2 შეჯამება (გამოყენებული მოდულები, პროგრამის მახასიათებლები და სპეციფიკაციები)22	
6.2.1 TSO მახასიათებლები და სპეციფიკაციები	22
6.3 უპირატესობები	23
7. დასკვნა	24
8. გამოყენებული ლიტერატურა	25

1. შესავალი

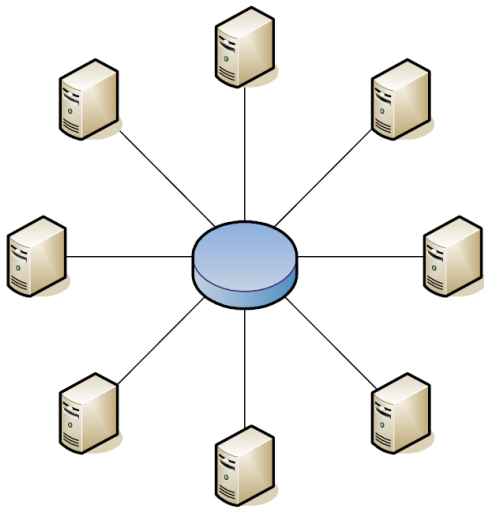
დღესდღეობით, ერთრანგოვანი ქსელის გამოყენებით შექმნილი პროგრამული უზრუნველყოფები სულ უფრო და უფრო პოპულარული ხდება. განსხვავებით გავრცელებული კლიენტ-სერვერული გადაწყვეტილებისგან, P2P-ს აქვს რამდენიმე უპირატესობა. მათ შორის: გაზრდილი სიმტკიცე და გაზრდილი რესურსები, როგორცაა: მეტი გამტარიანობა, მეტი სათავსო სივრცე და მეტი კომპიუტერული სიმძლავრე. სფერო, სადაც სიმტკიცე და რესურსების გამოყენება მნიშვნელოვანია, არის - განსაკუთრებით დიდი ფაილების გადანაწილება. მაგალითად, Peer-to-peer გადაწყვეტა, რომელიც წარმოადგენს კლასიკური სერვერ-კლიენტის გადაწყვეტის ეფექტურ და საიმედო ალტერნატივას, არის BitTorrent.

ნაშრომში ყურადღება გავამახვილდება BitTorrent-ზე, როგორც peer-to-peer გადაწყვეტაზე, მის არქიტექტურასა და ცნებებზე, რომლებიც შეადგენს BitTorrent-ს. ნაშრომში წარმოდგენილია და განიხილება BitTorrent-ის გამოყენების მოკლე მიმოხილვა, ისტორია და პერსპექტივები.

სამაგისტრო ნაშრომის ძირითად შედეგებად შეიძლება მივიჩნიოთ: ერთრანგოვანი ქსელის მიმოხილვა, BitTorrent-ის პროცესების ანალიზი, პროტოკოლში გამოყენებული ალგორითმების წარმოდგენა და მიღებული ცოდნის საფუძველზე, პითონის ასინქრონული მოდულის დახმარებით შექმნილი, სრულყოფილი, მულტიპლათფორმული ბიტტორენტ კლიენტი.

2. ერთრანგოვანი ქსელი (P2P)

მიუხედავად იმისა რომ ერთრანგოვანმა ქსელმა ყურადღება მიიპყრო ცოტა ხნის წინ, მუსიკასა და კინოინდუსტრიაში მიმდინარე ბრძოლის გამო (იმის გამო რომ, ბიტტორენტის საშუალებით ძირითად ვრცელდება პირატული მასალა, ბრენ კოენს, არაერთმა წამყვანმა მუსიკისა თუ კინოინდუსტრიის წამყვანმა კომპანიებმა უჩივლა), ის სულაც არაა ახალი კონცეფცია. ერთრანგოვანი ქსელი მარტივად შესაძლებელია განიმარტოს: კომუნიკაციის მოდელი, რომელშიც თითოეულ მხარეს აქვს ერთი იგივე შესაძლებლობები და ნებისმიერმა მხარემ შეიძლება დაიწყოს საკომუნიკაციო სესია [\[17\]](#).



ფიგურა 1 - კლიენტ-სერვერული მოდელი

ეს ნიშნავს, რომ კონცეპტუალურად, ერთრანგოვანი ქსელი (P2P) არის ალტერნატივა ტრადიციული კლიენტ-სერვერული არქიტექტურისა, სადაც ჩვეულებრივ არის ერთი სერვერი და მრავალი კლიენტი (ფიგურა 1) [\[23\]](#).

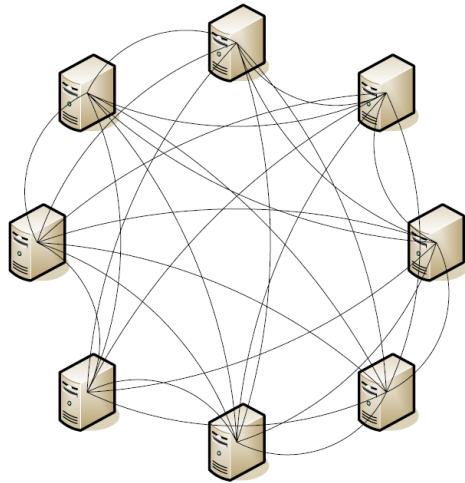
წინა განსაზღვრების თანახმად, ერთრანგოვანი ქსელის მაგალითად შეიძლება განვიხილოთ 1960-იან წლებში, ინტერნეტის ადრეული განხორციელება (ARPANET), რომელშიც მონაწილე ყველა მხარე იყო თანასწორი შესაძლებლობის მქონე. თუმცა, ეს განსაზღვრება შეიძლება ბევრნაირად იყოს გაგებული. დომენური სახელების სისტემა (DNS) არის კარგი მაგალითი იმისა თუ როგორაა შესაძლებელი რომ ის მივაკუთვნოთ, როგორც ტრადიციული ერთრანგოვან ქსელს, ისე ინფორმაციული საკუთრების იერარქიული მოდელსაც. ამიტომაც, ერთრანგოვანი ქსელის უფრო ზუსტი განსაზღვრება ჟღერს, შემდეგნაირად:

განაწილებული ქსელის არქიტექტურას შეიძლება ეწოდოს ერთრანგოვანი (P2P), თუ მასში მონაწილე მხარეები ინაწილებენ (საზიარო აქვთ) საკუთარი (აპარატურის) რესურსების ნაწილს და ეს საერთო რესურსები აუცილებელია ქსელის მიერ შემოთავაზებული სერვისისა და დანიშნულების უზრუნველსაყოფისთვის. რესურსები წვდომადია სხვა მონაწილეებისთვის, ყოველგვარი შუამავალი ობიექტების გარეშე. ასეთი ქსელის მონაწილე ნებისმიერი მხარე, თავადაა როგორც რესურსის პროვაიდერი, ისე რესურსის მომთხოვნიც [\[7\]](#) [\[17\]](#).

2.1 ერთრანგოვანი ქსელის ტოპოლოგიები

2.1.1 წმინდა ერთრანგოვანი ქსელი

წმინდა ერთრანგოვანი ქსელი არის ერთრანგოვანი ქსელი, რომელშიც ქსელის შიგნით ხელმისაწვდომი ერთადერი ობიექტები, თავად ქსელში მონაწილე მხარეები არიან.



ფიგურა 2 წმინდა ერთრანგოვანი ქსელი

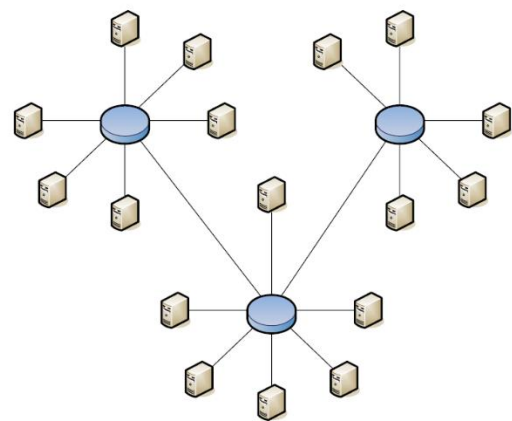
ასევე ის შეგვიძლია განვიხილოთ შემდეგნაირად: პირველ რიგში ის უნდა აკმაყოფილებდეს ერთრანგოვანი ქსელის დაზუსტებულ განმარტებასა და მეორე, თუ კი ქსელიდან შემთხვევითობის პრინციპით მოვაშორებთ რომელიმე მონაწილე მხარეს, ამით ქსელის მომსახურებას არანაირი ზიანი არ მიადგება და სერვისი შენარჩუნდება.

ეს კონცეფცია მოცემულია მეორე ფიგურაზე. მასზე კარგად ვხედავთ, რომ ქსელში მხოლოდ თანაბარი შესაძლებლობის მქონე მონაწილე მხარეებია წარმოდგენილი. ყველა ყველასთანაა დაკავშირებული, რაც ნიშნავს იმას, რომ ნებისმიერის ქსელიდან მოშორებას არ მოყვება ფატალური შედეგი ქსელზე.

2.1.2 ჰიბრიდული ერთრანგოვანი ქსელი

განსხვავებით, წმინდა ერთრანგოვანი ქსელისგან, სადაც მხოლოდ თავად მონაწილე მხარეებს შეეძლოთ ქსელში ყოფილიყვნენ, ჰიბრიდული ქსელი ყოველთვის მოიცავს გარკვეულ ცენტრალურ ერთეულს. ეს ნაჩვენებია მესამე სურათზე. ჰიბრიდული ერთგვაროვანი ქსელი განისაზღვრება, როგორც:

განაწილებული ქსელის არქიტექტურა კლასიფიცირდება ჰიბრიდულ ერთრანგოვან ქსელად, თუ კი პირველი, ისაა ერთრანგოვანი ქსელი მისი დაზუსტებული განსაზღვრების თანახმად და მეორე, აუცილებელია ცენტრალური ობიექტის არსებობა ქსელის უზრუნველსაყოფად.

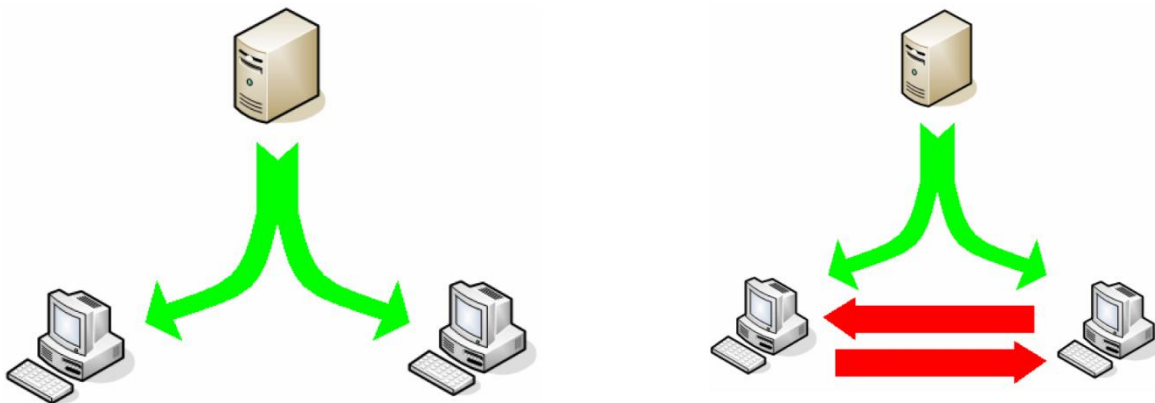


ფიგურა 3 ჰიბრიდული ერთრანგოვანი ქსელი

როგორც მესამე სურათიდან ჩანს, ახლა ქსელში გვაქვს ისეთი ობიექტები, რომელიც აკავშირებს სხვადასხვა ქსელს. თუ კი რომელიმე ასეთი ობიექტი მწყობრიდან გამოვა, მყისიერად ქსელის გარკვეული ნაწილები ერთმანეთისგან გამოიყოფა. ეს ყოველივე ჰიბრიდულ ერთრანგოვან ქსელს უფრო დაუცველს ხდის თავდასხმების წინააღმდეგ.

3. BitTorrent - შესავალი

BitTorrent არის ტექნოლოგია/პროტოკოლი, რომელიც გამოიყენება ფაილების გავრცელებაში, მათ შორის - განსაკუთრებით, დიდი ფაილების. ის გავრცელების პროცესს უფრო მარტივს ხდის და ფაილის საწყისი მფლობელის ქსელს ნაკლებად ტვირთავს. ეს ყოველივე მიიღწევა სხვა მომხმარებლების (peers) ხარჯზე, რომლებიც ფაილს იწერენ. რაც უფრო მეტი გადმოიწერი ყავს მოცემულ ფაილს, მით უფრო მცირდება დატვირთვა ფაილის თავდაპირველი მომხმარებლის ქსელზე.



ფიგურა 4 ბიტტორენტ პროტოკოლის ძირითადი მიმდინარეობა

მეოთხე სურათზე ნაჩვენებია, ბიტტორენტ პროტოკოლის ძირითადი მიმდინარეობა. სურათის მარცხენა ნაწილში წარმოდგენილია კლიენტ-სერვერული გადაწყვეტა. მომხმარებლები სერვერიდან იწერენ ერთდროულად. თუ ვივარაუდებთ, რომ სერვერის ატვირთვის მოცულობა, იგივეა რაც მომხმარებლის გადმოწერის მოცულობა, მაშინ თუ კი ერთი მომხმარებელი იწერს სერვერიდან, ფაილის ჩამოტვირთვას დაჭირდება ორჯერ მეტი დრო (ჯერ სერვერზე ატვირთვა, მერე ჩამოტვირთვა). მარჯვენა ნაწილში ნაჩვენებია ბიტტორენტის მსგავსი მიდგომა. ფაილის ორ ნაწილად გაყოფით თითოეული მომხმარებლისთვის, და დარჩენილი ნაწილის მეორე მომხმარებლისგან ჩამოტვირთვით მცირდება, როგორც მთლიანი ფაილის ჩამოტვირთვის დროც, ისე დატვირთვა სერვერზეც. რა თქმა უნდა ბიტტორენტი გაცილებით რთულია ვიდრე, მოცემული მარტივი მაგალითი, მაგრამ ის აღწერს მის ძირითად იდეას.

3.1 BitTorrent-ის ისტორია

BitTorrent უდავოდ ყველაზე პოპულარული ერთრანგოვანი პროგრამაა. სტატისტიკა მოწმობს, რომ მასზე მთლიანი ინტერნეტ ტრაფიკის 35%-ზე მეტი მოდის. როგორ გახდა ის ასე პოპულარული და რა ხდის მას განსაკუთრებულს?

90-იანი წლების ბოლოს, ბრემ კოენს (Bram Cohen) მობეზრდა ისეთ ინტერნეტ ბიზნეს კომპანიებში მუშაობა, რომლებიც არავითარ პროდუქტს არ უშვებდნენ და საბოლოოდ კოტრდებოდნენ კიდეც. ამიტომაც გადაწყვიტა დამოუკიდებლად შეექმნა პროდუქტი რისთვისაც იდეები, ბოლო კომპანიიდან მიიღო. წინა კომპანიის იდეა იყო, შეენახათ ფაილი მთლიანად და ნაწილობრივ. მისი პატარა ნაწილებად დაყოფით, მათი დაშიფვრითა და სხვადასხვა ადგილზე შენახვით. მან გაიაზრა, რომ ფაილის პატარა ნაწილებად დაყოფა, ასევე შესაძლებელი იყო გამოეყენებინა ფაილების გაზიარებისთვისაც [\[20\]](#).

2001 წლის ზაფხულში, კოენმა გამოუშვა ბიტტორენტის ბეტა ვერსია. 2002-ში კი ის კონფერენციაზე წარადგინა. მისი მიზანი იყო მომხმარებლისთვის მიეცა სწრაფი და მარტივი საშუალება ლინუქსის პროგრამული უზრუნველყოფის გავრცელებისა და მიმოცვლისთვის. მაგრამ, როგორც ყველამ ვიცით, ფილმების მოყვარულებმა მაშინვე დაინახეს დიდი პოტენციალი ბიტტორენტში. 2004-ში ბიტტორენტის ტრაფიკში პირატული ფილმებისა და ტელეშოუების მიმოცვლა გახდა დომინანტი და ამის შემდეგ ბიტტორენტი წარმოუდგენელი ტემპებით გაიზარდა. რამდენიმე წლის წინ ბიტტორენტის პროგრამის გადმოწერებმა 40 მილიონს გადააჭარბა. ბრემ კოენის ერთადერთი შემოსავალი არის ნებაყოფლობითი შემოწირულობა ბიტტორენტის კმაყოფილი მომხმარებლისგან [\[13\]](#).

ის სრულიად უფასოა და ღია კოდია (open-source). უეჭველია მის წარმატებაში ამ ფაქტორმა დიდი როლი ითამაშა, თუმცა ბიტტორენტი როგორც მუშაობს, ასევე საგრძნობლად განსხვავდება სხვა ერთრანგოვანი პროტოკოლებისგან.

ბრემ კოენის აზრით, „ტრადიციული“ ერთრანგოვანი პროტოკოლების პრობლემას წარმოადგენდა ის, რომ უმრავლეს მომხმარებელს ჰქონდა განსხვავებული ჩამოტვირთვისა და ატვირთვის სიჩქარე. ეს ნიშნავს იმას, რომ მიუხედავად იმისა, რომ შესაძლოა მომხმარებლის ქსელს, რომელიც იწერს ფაილს, გააჩნდეს საკმარისი გამტარუნარიანობა, მისი გადმოწერის სიჩქარე მაინც შეზღუდული იქნება, მეორე მომხმარებლის ატვირთვის სიჩქარით.

ეს კი ნიშნავს იმას, რომ ერთი ერთზე ფაილის გაზიარება არ არის ოპტიმალური გადაწყვეტა. არადა სწორედ ესაა „ტრადიციული“ ერთრანგოვანი პროტოკოლების მუშაობის მეთოდი (მაგალითად, KaZaa პროტოკოლი). ბრემ კოენმა, ეს პრობლემა,

ფაილის პატარ-პატარა ნაწილებად დაყოფით გადაჭრა. ფაილის მოთხოვნისას, მომხმარებლის კომპიუტერი ეძებს ინტერნეტში იმ ხალხს, რომელთაც ფაილის ერთი ან მეტი ნაწილი აქვთ. შემდეგ, იგი ერთდროულად იწერს ფაილის სხვადასხვა ნაწილს სხვადასხვა მომხმარებლისგან და უკეთესად იყენებს ქსელის გამტარუნარიანობას. მოც. გადაწყვეტის მთავარი ეფექტი არის რომ, ფაილი გაცილებით მალე ჩამოიწერება, ვიდრე როცა ფაილის გადმოწერა ხდება მხოლოდ ერთი მომხმარებლისგან. მარტივად ჟღერს, არა?! მომდევნო თავებში განვიხილავ, თუ როგორ ხდის ამ ყოველივეს ბიტტორენტი შესაძლებელს.

3.2 გამოყენების სფეროები

პირველი რაც შეიძლება მოგვაფიქრდეს, მას შემდეგ რაც გვესმის ერთრანგოვანი ქსელი არის ის რომ ბიტტორენტი ასოცირდება მეკობრეობასა და არალეგალურ გავრცელებასთან. ეს სულაც არაა აზრს მოკლებული, მართლაც ტრაფიკის უმეტეს ნაწილს საავტორო უფლებებით დაცული ფაილების არალეგალურად გავრცელება შეადგენს. მაგრამ ბიტტორენტ პროტოკოლს აქვს ფუნქციები, რომლებიც მას სავსებით გამოსადეგს ხდის კანონიერი მიზნებისთვის.

უკვე შესაძლებელია ლინუქს დისტრიბუტივების ჩამოტვირთვა ბიტტორენტის საშუალებით, რაც გაცილებით უფრო სწრაფია ვიდრე რეგულარული FTP ან HTTP პროტოკოლები. როგორც წინ აღვნიშნეთ ეს იყო ერთერთი მიზეზი რატომაც ბრემ კონმა შემქნა ბიტტორენტ პროტოკოლი. მაგალითისთვის, ვებ ბრაუზერი Opera-ც შესაძლებელია გადმოიწეროს ბიტტორენტის საშუალებით [16]. ეს არამარტო ზრდის გადმოწერის სიჩქარეს, ასევე ამცირებს სერვერებზე დატვირთვას, ყოველი ახალი ვერსიის გამოშვებისას. მას შემდეგ რაც მათ გამოუშვეს ბრაუზერის მერვე ვერსია, მათი სერვერები მწყობრიდან გამოვიდა დიდი მოთხოვნის გამო [14] [15]. პროტოკოლის ლეგალური მიზნებისთვის გამოყენება დღითიდღე იზრდება, ბევრი იყენებს მას თავისი მუსიკისა თუ მოკლემეტრაჟიანი ფილმების გასაზიარებლად, თამაშების დემო ვერსიების გასავრცელებლად და სხვა [9].

ბიტტორენტი ასევე გამოიყენება ბიზნესში. მისი დახმარებით შესაძლებელია მაღალი სიჩქარით გავრცელდეს ISO ფაილები, ოპერაციული სისტემები, დიდი ზომის პროგრამული უზრუნველყოფები და პროგრამული განახლებები. მაგალითისთვის, ყველას გამოუცდია თუ რამდენ ხანს უნდება ვინდოუსის უსაფრთხოების განახლების გადმოწერა, ხოლო ბიტტორენტის გამოყენებით მსგავსი ტიპის პროგრამული უზრუნველყოფების გავრცელება შესაძლებელია უფრო სწრაფად და შესაბამისად ისე რომ, კლიენტი გაცილებით კმაყოფილი დარჩეს. თვითონ ორგანიზაციების შიგნითაც შესაძლებელია პროტოკოლი გამოყენებულ იქნას პროგრამების და განახლებების გავრცელებისთვის.

ბიტტორენტი ყველაზე მეტად შეესაბამება ახალ, პოპულარულ ფაილებს, რომლებიც აქტუალური და საინტერესოა მომხმარებელთა უმეტესი ნაწილისთვის. მომხმარებლისთვის მარტივია იპოვოს ფაილის სხვადასხვა ნაწილი და გადმოიწეროს ისინი სწრაფად. ეს ეფექტი შეგვიძლია მოვიხსენიოთ როგორც მულტიპლიკატორის ეფექტად და მაგალითისთვის ნაკლებად პოპულარული ფილმი ანდა სატელევიზიო შოუ შესაძლებელია ძალიან პოპულარული გახდეს, რამდენიმე დღეში, ანდა საათებშიც კი. ხოლო ძველი და ნაკლებად პოპულარული ფაილები ძნელად საპოვნელი იქნება და თან, მხოლოდ რამდენიმე მომხმარებელი თუ მოიძებნება ვისაც ეს ფაილები აქვს.

4. BitTorrent არქიტექტურა

ბიტტორენტ არქიტექტურა ძირითადად მოიცავს შემდეგ ნაწილებს [\[27\]](#):

- სტატიკურ მეტაინფო ფაილი (ე.წ. ტორენტ ფაილი)
- ტრეკერი (a tracker)
- ორიგინალი გადმომწერი („seed“)
- მოცემულ მომენტში ვინც იწერენ ის მომხმარებლები („leecher“)

პირველი ნაბიჯი იმისთვის რომ ბიტტორენტის გამოყენებით ფაილი გამოვაქვეყნოთ, საჭიროა მოცემული ფაილის მეტაინფო ფაილი შევქმანათ (ე.წ. ტორენტ ფაილი). ტორენტ ფაილი შეიცავს ფაილის სახელს, ზომას, შიგთავსს, ინფორმაციას როგორაა დაშიფრული ფაილი და ტრეკერის მისამართს. ტორენტ ფაილები სჭირდება ყველას ვისაც სურს მოც. ფაილის გადმოწერა.

ტორენტ ფაილები იქმნება უფასო პროგრამების გამოყენებით. მოც. ფუნქციონალი უმეტესწილად აქვს თავად ტორენტ კლიენტებს. იმისთვის რომ გადმოვიწეროთ ტორენტის გამოყენებით ფაილი, საჭიროა გვეყენოს ტორენტ კლიენტი [\[28\]](#). ბიტტორენტ კლიენტი არის პროგრამული უზრუნველყოფა, რომელიც ადმინისტრირებას უწევს გადმოწერის პროცესს. გადმოწერა იწყება მას შემდეგ რაც ტორენტ ფაილს გავხსნით ტორენტ კლიენტის საშუალებებით [\[29\]](#).

ტრეკერი აწარმოებს ლოგირებას: თუ რომელი მომხმარებლები იწერენ ფაილს და ეხმარება მათ იპოვონ ერთმანეთი. ტრეკერი თავად არ იღებს მონაწილეობას ინფორმაციის გადაწოდებაში და არ აქვს ფაილის კოპია. ტრეკერი და გადმომწერი მომხმარებლები ინფორმაციას ცვლიან HTTP პროტოკოლით. თავიდან მომხმარებელი ტრეკერს ატყობინებს რომელ ფაილს იწერს, რომელ პორტებს უსმენს და ა.შ. ტრეკერის პასუხი არის სხვა მომხმარებლების სია, ვინც მოც. ფაილს იწერენ და ინფორმაცია თუ

როგორ მოხდეს მათთან კომუნიკაცია. იმ მომხმარებლების ჯგუფს რომლებიც იზიარებენ ერთსა და იმავე ტორენტს უწოდებენ გუნდს („swarm“).

მიუხედავად ამისა, მარტო ტორენტ ფაილის შექმნა არ არის საკმარისი. საჭიროა ფაილის საწყისმა მფლობელმა ატვირთვა დაიწყოს. ამ მომხმარებელს სიდერს უწოდებენ, ხოლო გადმომწერ მომხმარებლებს რომელთაც ჯერ არაფერი გადმოუწერიათ ანდა ფაილის ნაწილი აქვთ ჯერ გადმოწერილი leecher-ად მოიხსენიებენ. Seed-მა უნდა ატვირთოს ერთი მთლიანი კოპია მაინც. მას შემდეგ რაც კოპია გავრცელდება სხვა გადმომწერეს შორის, სიდერს შეუძლია გათიშოს ფაილის ატვირთვა და მიუხედავად ამისა ფაილის გადმოწერა გაგრძელდება ჩვეულებრივ - უკვე საკმარის მომხმარებელს აქვს გადმოწერილი. პოპულარული ფაილისთვის ერთი სრული კოპია საკმარისია, მაგრამ ნაკლებად ცნობილი ფაილისთვის შესაძლოა საჭირო იყოს სიდერმა კვლავ გაარძელოს ფაილის ატვირთვა ერთი მთლიანი კოპიის შემდეგაც. რაც მეტი გადმომწერი ყავს ფაილს, მით მეტად მცირდება საწყისი სიდერის დატვირთვა. ტრეკერის დატვირთვაც მცირეა, მიუხედავად იმისა რომ გადმოწერების რაოდენობა ძალიან მაღალია [10].

როდესაც იქმნება ტორენტ ფაილი ორიგინალი ფაილიდან, ორიგინალი ფაილი ნაწევრდება 512 ან 256 კილობაიტ ზომის ნაწილებად. ნაწილების SHA-1 ჰეშკოდებიც არის ტორენტ ფაილში. გადმოწერილი ინფორმაცია მოწმდება, მასზე SHA-1 ჰეშის გამოთვლით და ის დარდება ტორენტ ფაილიდან შესაბამისი ნაწილის SHA-1 ჰეშს. ამ მეთოდით თავიდან ვირიდებთ შეცდომას და კლიენტი გარანტირებულად ღებულობს ნამდვილ ფაილს. მას შემდეგ რაც რომელიმე ნაწილი გადმოიწერება და შემოწმდება, სიდერი ატყობინებს სხვა სიდერებს მოც. ნაწილის შესახებ.

4.1 ალგორითმები

ბევრ ერთრანგოვან პროტოკოლებში ფაილების გაცვლა ხდება ერთი ერთზე, რაც ნიშნავს იმას რომ მომხმარებელი თავად ირჩევს ვისგან გადმოწეროს ფაილი. როგორც ზემოთ აღვწერეთ, BitTorrent- ის კონცეფციაა ერთდროულად შეიძლებოდეს ბევრი სხვა მომხმარებლისგან გადმოტვირთვა. ეს მოითხოვს ვიცოდეთ რომელი მომხმარებლისგან რომელი ნაწილი გადმოვიწეროთ, რომ რაც შეიძლება სწრაფად მივიღოთ სრული ფაილი. ეს უნდა მოხდეს საბოლოო მომხმარებლის ჩართულობის გარეშე და უნდა იყოს თვითკონფიგურებადი.

მომხმარებლის შერჩევა თუ ვისგან უნდა გადმოვიწეროთ ფაილი აირს ორმხრივი პრობლემა. პირველი, უნდა ვიცოდეთ ნაწილების ჩამოტვირთვის საუკეთესო თანმიმდევრობა. ეს განისაზღვრება ნაწილის შერჩევის ალგორითმით. მაგრამ ეს არაა საკმარისი. მომხმარებელმა, რომელსაც აქვს თქვენთვის სასურველი ნაწილი, შესაძლოა

ის არ გადმოგაწერინოთ. ის სტრატეგიები რომელიც ამის საშუალებას აძლევს მოხმარებელს ცნობილია როგორც choking (დახრჩობა) და ეხება რესურსების გამოყოფას.

4.1.1 ნაწილის შერჩევის ალგორითმი

თუ რომელი ნაწილის გადმოწერას ირჩევს BitTorrent დიდ გავლენას ახდენს პროტოკოლის შესრულებაზე. საჭიროა ჭკვიანურად შეირჩეს თუ რომელი ნაწილი გადმოიწეროს, რადგან არ მოხდეს ისე რომ ყველა მომხმარებელს ქონდეს მხოლოდ აქამდე არსებული ფაილის ნაწილები და არავის - დარჩენილი ნაწილი.

მიზანი არის ის რომ რაც შეიძლება სწრაფად მოხდეს ნაწილების კოპირება სხვადასხვა მომხმარებელში. ეს გაზრდის ჩამოტვირთვის სიჩქარეს და ასევე დარწმუნებული ვიქნებით რომ, ფაილის ყველა ნაწილია ქსელში, იმ შემთხვევაში თუ სიდერი შეწყვეტს ატვირთვას.

4.1.2 ქვენაწილები

BitTorrent იყენებს TCP-ს და ამიტომაც მნიშვნელოვანია მუდმივი მონაცემების გადაცემა, თუ არადა გადაცემის სიჩქარე დაეცემა ნელი სასტარტო მექანიზმის გამო. თითო ნაწილი კვლავ დაყოფილია უფრო პატარა ნაწილებად, ხშირად დაახლოებით 16kb ზომის. პროტოკოლი ყოველთვის დარწმუნებულია, რომ ყოველთვის აქვს გარკვეული რაოდენობის ქვენაწილის მოთხოვნა (როგორც წესი, ხუთი) ნებისმიერ დროს. როდესაც ახალი ქვენაწილი გადმოწერილია, ახალი მოთხოვნა გაიგზავნება. ქვენაწილები შეიძლება გადმოწერილი იყოს სხვადასხვა მომხმარებლისგან.

4.1.3 Policy #1: Strict Policy

მას შემდეგ რაც რომელიმე ქვენაწილზე ხდება მოთხოვნა, იგივე ნაწილის დარჩენილი ქვენაწილები მოითხოვება სხვა ყველა ნაწილის ქვენაწილამდე [\[12\]](#).

4.1.4 Policy #2: Rarest First

მთავარი წესი BitTorrent-ში არის „იშვითი პირველი“. ეს ნიშნავს იმას რომ როდესაც მომხმარებელი ირჩევს შემდეგ ნაწილს გადმოსაწერად, შეირჩევა ის ნაწილი რომელიც ყველაზე ცოტა სხვა მომხმარებელს აქვს [\[12\]](#).

4.1.5 Policy #3: Random First Piece

როდესაც ვიწყებთ გადმოწერას, ჯერ არ გვაქვს არცერთი ნაწილი რომ დავიწყოთ მათი გაზიარება. მნიშვნელოვანია რომ პირველი ნაწილი მივიღოთ რაც შეიძლება სწრაფად, ხოლო აქ მეორე წესი გამოუსადეგარია. თუ პირველ ნაწილად შევარჩევთ ყველაზე იშვიათ ნაწილს, გამოდის რომ ის გადმოიწერება ყველაზე ნელა, რადგან ის ისედაც ცოტა მომხმარებელს აქვს. სხვადასხვა მომხმარებელი ერთი და იმავე ნაწილით ზრდიან გადმოწერის სიჩქარეს, ამიტომაც ეს მესამე წესი გამოიყენება პირველი

ნაწილის შემთხვევითად შესარჩევად, რის შემდეგაც მეორე ნაწილის გადმოწერის დაწყებიდან ვუბრუნდებით მეორე წესს [\[12\]](#).

4.1.6 Policy #4: Endgame mode

ხანდახან რაიმე ნაწილის გადმოწერა შეიძლება ხდებოდეს ისეთი მომხმარებლისგან რომელსაც დაბალი ატვირთვის სიჩქარე აქვს, რამაც შესაძლოა შეაფერხოს გადმოწერის დასრულება. ამ მდგომარეობის პრევენციისთვის არსებობს ეს მეოთხე წესი. როგორც წინ აღვნიშნეთ სულ გვაქვს 5 ქვენაწილის მოთხოვნა, როდესაც კონკრეტულ მომხმარებელთან 5-ვე ქვენაწილიდან რომელიმეზე მოდის უარყოფითი პასუხი, მაშინ მოთხოვნა ეგზავნება მთელ ერთრანგოვან ქსელს. ხოლო როგორც კი ვღებულობთ ბოლო ნაწილს, იგზავნება მოთხოვნის გაუქმების მესიჯი, რომ ნაწილი უკვე მივიღეთ. რა თქმა უნდა ამ დროს რაღაც დატვირთვის ვახორციელებთ ზედმეტად, მაგრამ პრაქტიკაში ეს არ იგრძნობა, რადგანაც ამ მდგომარეობაში მოხმარებელს ან საერთოდ არ უწევს ყოფნა, ანდაც ძალიან ცოტა ხანია [\[12\]](#).

4.2 რესურსების გამოყოფა

BitTorrent- ში არ არსებობს ცენტრალიზებული რესურსების განაწილება. ყველა მომხმარებელი პასუხისმგებელია გადმოწერის სიჩქარეზე. ბუნებრივია, მომხმარებელი ცდილობს ჩამოტვირთოს ყველასგან, ვისგანაც შეუძლია [\[11\]](#).

4.2.1 Choking ალგორითმი

Choking ესაა დროებითი უარი ატვირთვაზე სხვა მომხმარებლისთვის, მაგრამ ამ მომხმარებელს მაინც შეუიძლია მისგან გადმოწერა. იმისთვის რომ მომხმარებლებმა ერთმანეთთან ითანამშრომლონ, ერთმანეთს რთავენ ნებას რომ ატვირთონ მისგან ფაილი, ხოლო თუ გადაწყვეტენ რომ არ ითანამშრომლონ მაშინ, ისინი ახშობენ სხვა მომხმარებლების კავშირს. პრინციპი მდგომარეობს იმაში რომ, შენ ტვირთავ იმ მომხმარებლისთვის, რომლებმაც ცოტა ხნის წინ აგიტვირთეს ფაილის ნაწილი. მიზანია რომ მუდმივად არსებობდეს რამდენიმე ორმხრივი კავშირი.

ამრიგად მთავარი კითხვა ახლა არის ის რომ თუ რომელი მომხმარებელი „დაახრჩონ“ და რომელს გაანებონ თავი. პროტოკოლის შიგნით მომხმარებელი „unchoke“-ს უკეთებს ფიქსირებული რაოდენობის სხვა მომხმარებელს. თუ ვინ უნდა აირჩეს მათ შორის, ამოირჩევა მათი გადმოწერის სიჩქარით. მომხმარებლების სიჩქარე დაითვლება 20 წამიან შუალედში [\[19\]](#).

შედეგად ვიღებთ იმას, რომ ნებისმიერი მომხმარებელი ატვირთავს იმ მომხმარებლისთვის, რომელთაც აქვთ მაღალი გადმოწერის სიჩქარე. სხვა მხრივ, თუ რომელიმე მომხმარებლის ატვირთვის სიჩქარე მაღალია, ასეთ მომხმარებელს უფრო

მეტი მომხმარებელი რთავს ნებას, რომ მისგან გადმოიწეროს. ეს ნიშნავს იმას რომ მიიღებთ მაღალ გადმოწერის სიჩქარეს თუ ბევრი ატვირთვებიც გაქვთ. ესაა ბიტტორენტ პროტოკოლის ერთერთი ყველაზე მნიშვნელოვანი მახასიათებელი. ამ გზით თავიდან ვიცილებთ ისეთ მომხმარებლებს რომლებიც მხოლოდ იწერენ და თავად არ ტვირთავენ. უმრავლეს ერთრანგოვან პროტოკოლებში ეს შეზღუდვა არ გხვდება, ამიტომაცაა ბიტტორენტი ასეთი პოპულარული.

4.2.2 ოპტიმისტური „unchoking“

ბიტტორენტი ასევე „unchoke“-ს უკეთებს რომელიმე მომხმარებელს, რომლის შერჩევაშიც მისი გადმოწერის სიჩქარე არ გაითვალისწინება. ამის მიზეზია ის რომ არსებობს გამოუყენებელი კავშირები და შესაძლოა ამ მომხმარებლიდან რომელიმე უკეთესია ვიდრე ვისთანაც გვაქვს კომუნიკაცია. ასეთი შერჩევა ყოველ 30 წამში ხდება. და თუ კი რომელიმე მომხმარებელი უფრო მეტ გადმოწერის სიჩქარეს გვთავაზობს, მაშინ ეს ჩაანაცვლებს არჩეულიდან ერთერთს [5].

4.2.3 Anti-snubbing

რა მოუვა მომხმარებელს, თუ უეცრად ყველა ის მომხმარებელი ვისგანაც ის იწერდა „დაახრჩობს“ კავშირს? ასეთ დროს უნდა ვიპოვოთ ახალი მომხმარებლები, მაგრამ ოპტიმისტური „unchoking“ ყოველ 30 წამში ერთხელ ეშვება და ისიც მხოლოდ 1 კავშირს პოულობს. ამისგან თავის ასაცილებლად, თუ კი რომელიმე მომხმარებლისგან 60 წამის განმავლობაში არაფერი მიგვიღია, მაშინ ვთვლით რომ ასეთი მომხმარებელი გვაიგნორებს და მასთან ვწყვეტთ ატვირთვას [3]. ამ რიგად ოპტიმისტური „unchoking“ რაოდენობა იზრდება და ამ გზით ვპოულობთ ახალ კავშირებს [4].

4.2.4 მხოლოდ ატვირთვა

როგორც ვნახეთ, მოც. ალგორითმით თუ მე შემიძლია მათგან სწრაფად გადმოვიწერო, მაშინ მათაც შეუძლიათ ჩემგან ატვირთონ. მაგრამ რა ხდება მაშინ როცა არაფერი მაქვს გადმოსაწერი? როგორ უნდა შევარჩიო ის მომხმარებლები, რომლებსაც ნება უნდა დავრთო ჩემგან ატვირთონ? როცა გადმოწერა სრულდება, გამოიყენება ახალი ალგორითმი, რომლებიც ნებას რთავს ჩვენგან ატვირთონ იმ მომხმარებლებმა, რომელთაც ატვირთვა ყველაზე სწრაფად შეუძლიათ. ამ გზით ფაილის ნაწილები აიტვირთვება სწრაფად და მათი კოპიოებიც მალე გამრავლდება.

4.2.5 დეცენტრალიზებული ტრეკერი

ერთი ცენტრალიზებული ტრეკერით ბიტტორენტ ქსელი არამდგრადია. თუ ტრეკერი გამოვა მწყობრიდან, ფაილი აღარ იქნება ხელმისაწვდომი, ტრეკერის გარეშე მომხმარებლებმა ერთმანეთი არ იციან. 2005 წელს, ბრემ კონენმა გამოუშვა პროტოკოლის მეოთხე ვერსია, რომელიც მხარს უჭერს დეცენტრალიზებულ ტრეკერს. ამ ვერსიაშია საკმარისია მხოლოდ გამოაქვეყნოთ ტორენტ ფაილი და დეცენტრალიზებულ ტრეკერში ქსელში არსებული თითოეული წევრი იქცევა ისე რომ

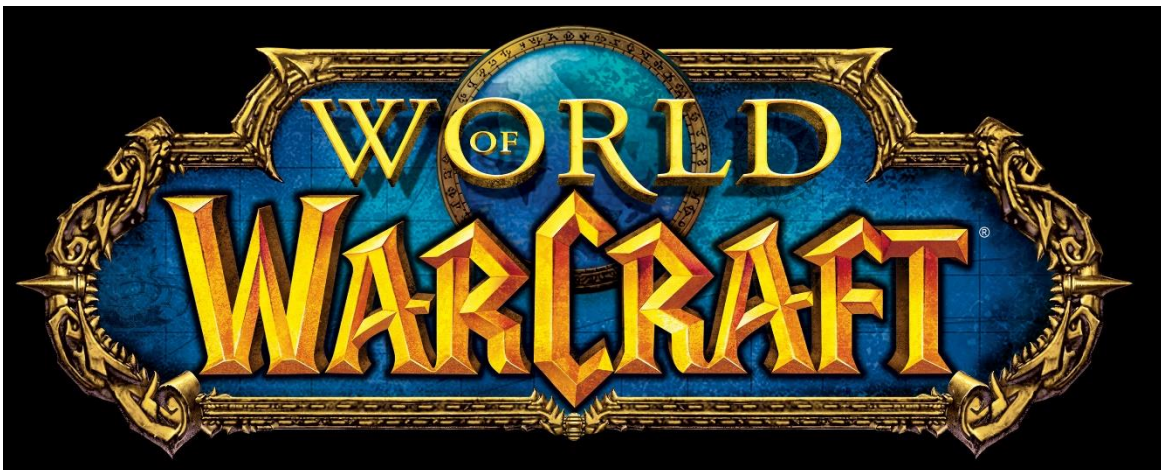
თითქოს თავადაა ცენტრალიზებული ტრეკერი [6]. დეცენტრალიზაციის გადაწყვეტა ხდება განაწილებული ჰეშ ცხრილების გამოყენებით [26] (distributed hash tables - DHT) [24] [25].

5. ბიტკორენტის პერსპექტივები

ბიტკორენტის მომხმარებლები დღითიდღე მატულობს და ყოველ წამს გიგაბაიტობით ფაილები მიმოიცივლება [1]. ეს ფაქტი და დოკუმენტში აღწერილი მუშაობის პრინციპი მართალთაგ გვამღევს საშუალებას ვივარაუდოთ, რომ ბიტკორენტი არის ტექნოლოგია რომელსაც შეუძლია შეცვალოს ლანდშაფტი სამაუწყებლო ინდუსტრიასა და ფაილების მიმოცვლაში. მიუხედავად სხვადასხვა დავებისა ბრემ კოენის მიმართ, პროტოკოლი მუდმივად ვითარდება და დღითიდღე იხვეწება [2] [8] [18].

უკვე სახეზე გვაქვს აპლიკაციები რომლებიც RSS და ბიტკორენტის ნაზავია. მოცემული საშუალებას გვამღევს რსს ფაილებიდან გამომდინარე, მოვძებნოთ მაგალითად რომელიმე კონკრეტული შოუს სერიები. ბიტკორენტს აქვს საშუალება ინტერნეტი ერთ დიდ TiVo-დ აქციოს (ტივო მედიაბოქსია რომელიც საშუალებას გვამღევს ტელევიზიაში გასული გადაცემები ჩაიწეროს, გარკვეული ქვიორდ სიტყვების ირგვლივ, მაგ.: ყველა გადაცემა რომელიც ცხოველებს ეხება, ყველა ფილმი რომელშიც თამაშობს ესა თუ ის მსახიობი და სხვა) [21] [22].

ბიტკორენტი არის ბიზნეს ფაილების გავრცელების იაფი და საიმედო ალტერნატივა. რაც საკმაოდ დიდ დანაზოგებს იძლევა. მაგალითისთვის, Blizzard Entertainment უკვე იყენებს ბიტკორენტს თავისი ონლაინ თამაშის The World of Warcraft განახლებების გასავრცელებლად



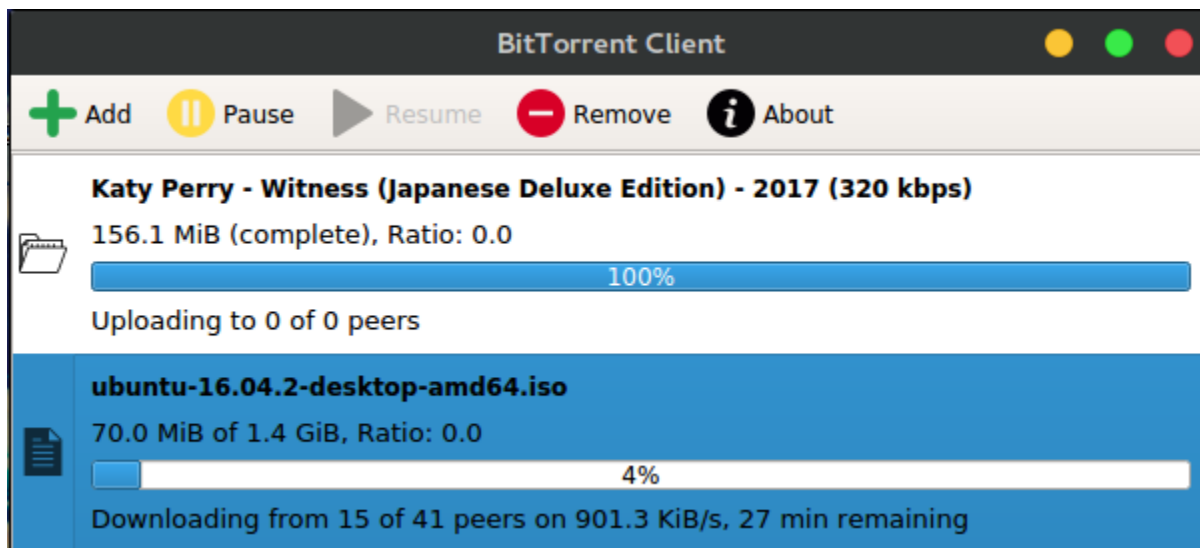
ფიგურა 5 The World of Warcraft-ის ლოგო

6. პროგრამული უზრუნველყოფა

მოცემულ ნაწილში წარმოდგენილია სამაგისტრო ნაშრომის შედეგი - მულტიპლატფორმული, სრულყოფილი, BitTorrent პროტოკოლთან თავსებადი, კლიენტ პროგრამული უზრუნველყოფა, პროგრამირების ენა, პითონის ასინქრონული ბიბლიოთეკის გამოყენებით - asyncio. პროგრამა წარმოდგენილია როგორც გრაფიკული ინტერფეისით, ასევე ავტორის მიერ შექმნილი პროგრამის გამოყენება შესაძლებელია როგორც command line (cli) ბრძანებად.

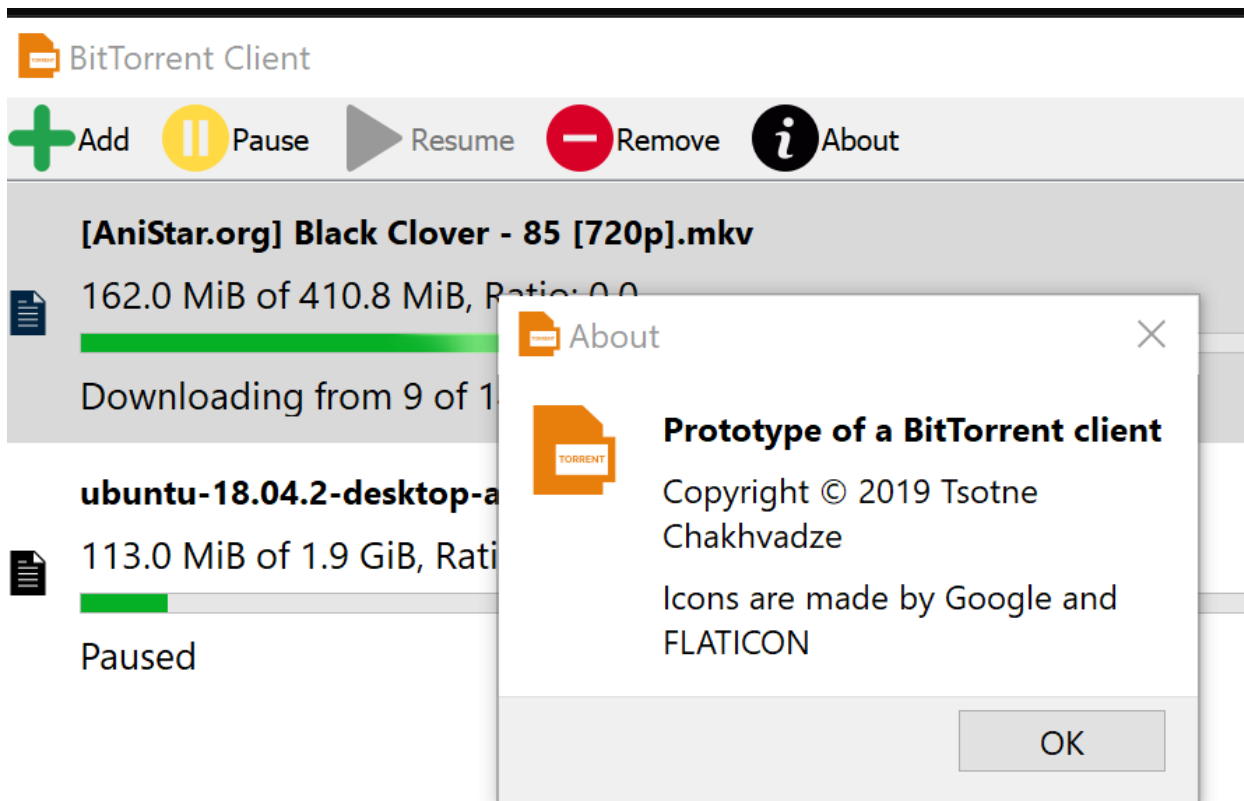
წარმოგიდგინოთ მიღებულ შედეგს:

მეექვსე სკრინზე წარმოდგენილია პროგრამის გრაფიკული ინტერფეისი, ნაჩვენებია ერთი აქტიური და ერთიც დასრულებული ტორენტ გადმოწერა.



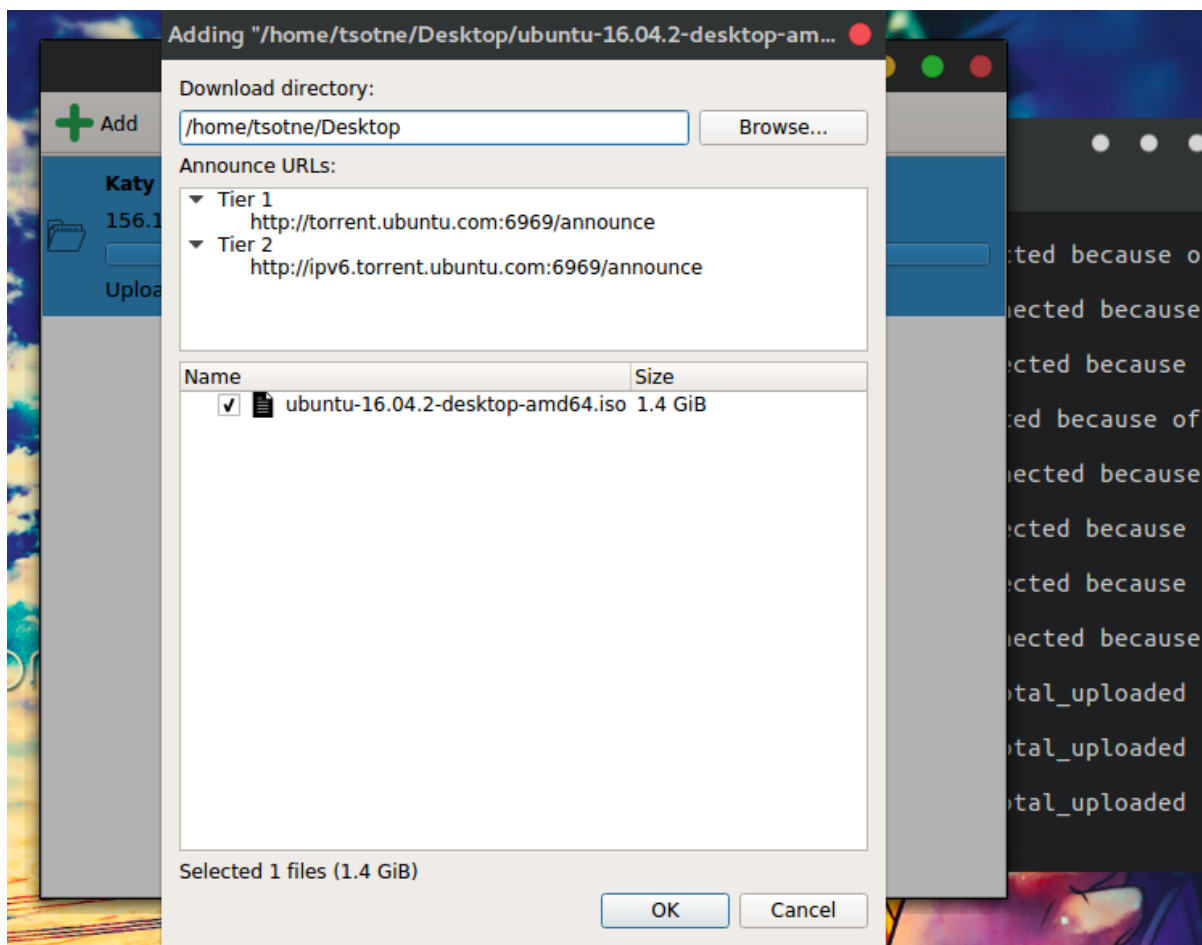
ფიგურა 6 პროგრამის გრაფიკული ინტერფეისი

მეშვიდე სკრინზე ნაჩვენებია პროგრამის შესახებ არსებული განყოფილება - იგი აღწერს, რომ პროგრამული უზრუნველყოფა წარმოადგენს ბიტტორენტის კლიენტ პროგრამას და ის შექმნილია ავტორის მიერ 2019 წელს. პროგრამაში გამოყენებული აიკონები ნაპოვნია „google“-ის გამოყენებით და ეკუთვნის Flaticon-ს.



ფიგურა 7 პროგრამის შესახებ განყოფილება

მომდევნო სურათი აღწერს, პროგრამაში ახალი ტორენტ ფაილის გახსნას. სურათზე ნაჩვენებია, რომ ყოველი ტორენტ ფაილის გახსნისას ვხედავთ ტრეკერის მისამართებს, ფაილში არსებულ კონტენტს (რომლიდანაც შესალებელია ამოირჩიოს მომხმარებელმა რომელი ფაილი გადმოიწეროს და რომელი არა) და თითოეული ფაილის ზომას. მომხმარებელს საშუალება აქვს თავად მიუთითოს გადმოსაწერი ფაილის შესანახი დირექტორია.



ფიგურა 8 ახალი ტორენტ ფაილის დამატება

6.1 BitTorrent კლიენტის კონსოლური ინტერფეისის გამოყენების ბრძანებები:

```
tstotne@tso:~/Desktop/TSO$ ./TSO_cli.py --help
usage: TSO_cli.py [-h] [--debug] ACTION ...
```

A prototype of BitTorrent client (console management tool)

optional arguments:

- h, --help show this help message and exit
- debug Show debug messages

subcommands:

Specify an action before "--help" to show parameters for it.

ACTION

start Start a daemon
stop Stop the daemon
show Show torrent content (no daemon required)
add Add a new torrent
pause Pause torrent
resume Resume torrent
remove Remove torrent
status Show status

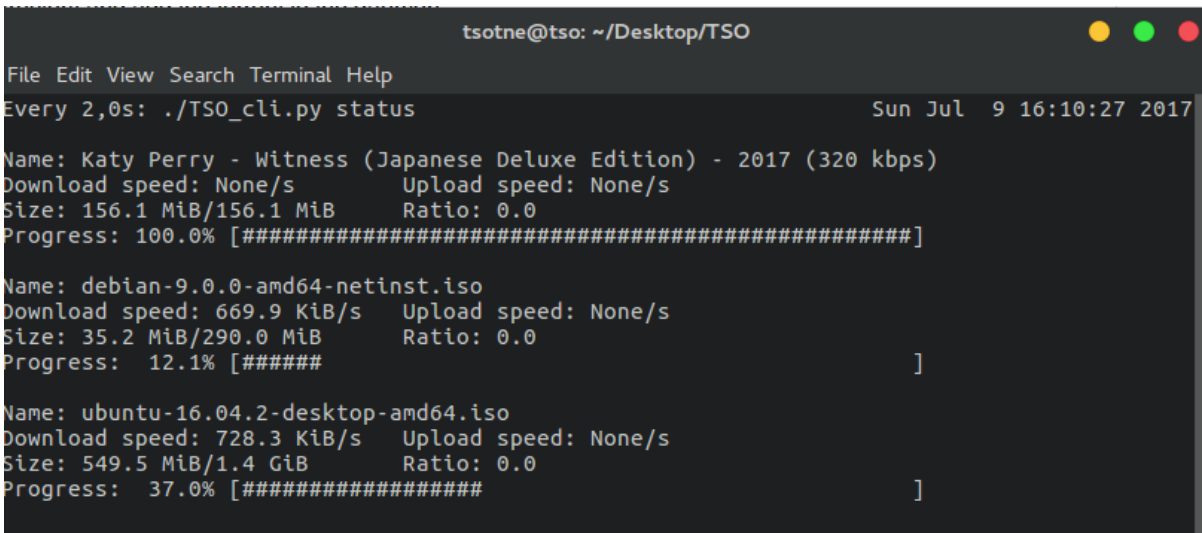
სქრინზე წარმოდგენილია, კონსოლურ ინტერფეისში ტორენტ ფაილის შიგთავსის ნახვა. გრაფიკული ინტერფეისის მსგავსად კონსოლურ ინტერფეისშიც წარმოდგენილია ტრეკერის მისამართი, ფაილის კონტენტი და თითოეული ფაილის ზომა.



```
tsozne@tso: ~/Desktop/TSO
File Edit View Search Terminal Help
tsozne@tso:~/Desktop/TSO$ ./TSO_cli.py show ../debian-9.0.0-amd64-netinst.iso.torrent
Name: debian-9.0.0-amd64-netinst.iso
ID: d4abefdf19c5a9ab73ced389faca97bdcbb2ef3f
Announce URLs:
  Tier 1: http://bttracker.debian.org:6969/announce
Content: single file (290.0 MiB)
tsozne@tso:~/Desktop/TSO$
```

ფიგურა 9 კონსოლური ინტერფეისი - ტორენტ ფაილის შიგთავსის ნახვა

მე-10 სქრინზე წარმოდგენილია კონსოლურ ინტერფეისში, პროგრამაში გახსნილი ტორენტ ფაილების სტატუსი.



```
tsozne@tso: ~/Desktop/TSO
File Edit View Search Terminal Help
Every 2,0s: ./TSO_cli.py status Sun Jul 9 16:10:27 2017
Name: Katy Perry - Witness (Japanese Deluxe Edition) - 2017 (320 kbps)
Download speed: None/s Upload speed: None/s
Size: 156.1 MiB/156.1 MiB Ratio: 0.0
Progress: 100.0% [#####]
Name: debian-9.0.0-amd64-netinst.iso
Download speed: 669.9 KiB/s Upload speed: None/s
Size: 35.2 MiB/290.0 MiB Ratio: 0.0
Progress: 12.1% [##### ]
Name: ubuntu-16.04.2-desktop-amd64.iso
Download speed: 728.3 KiB/s Upload speed: None/s
Size: 549.5 MiB/1.4 GiB Ratio: 0.0
Progress: 37.0% [##### ]
```

ფიგურა 10 ტორენტ ფაილების სტატუსი

მოცემული სკრინი ასახავს, ტორნეტ ფაილის გადმოწერის გაგრძელებასა და დაპაუზებას.

```
tsozne@tso: ~/Desktop/TSO
File Edit View Search Terminal Help
tsozne@tso:~/Desktop/TSO$ ./TSO_cli.py --debug resume ../ubuntu-16.04.2-desktop-amd64.iso.torrent
tsozne@tso:~/Desktop/TSO$ ./TSO_cli.py --debug pause ../ubuntu-16.04.2-desktop-amd64.iso.torrent
tsozne@tso:~/Desktop/TSO$
```

ფიგურა 11 გადმოწერის გაგრძელება და დაპაუზება

პროექტის სიმბიმიდან და სიდიდიდან გამომდინარე, უზრუნველყოფა მხარდაჭერილია ე.წ. ლოგირებით :

```
tsozne@tso: ~/Desktop/TSO
File Edit View Search Terminal Help
tsozne@tso:~/Desktop/TSO$ ./TSO_gui.py --debug
INFO 15:54:45 client_impl.net.peer_tcp_server server started on port 6881
INFO 15:54:45 client_impl.control.server server started on port 6995
INFO 15:54:45 client_impl.control.manage state recovered (1 torrents)
INFO 15:54:45 client_impl.net.tracker_clients.http announcing to http://bt2.t-ru.org/ann
INFO 15:54:45 client_impl.net.tracker_clients.http url_enc= http://bt2.t-ru.org/ann?compact=1&left=0&info_hash=%2CzP%82%7D%C3J%11%EDI%7C%FDzE%7B%EA%23%F2t&downloaded=163763027&peer_id=%21-%14%B2%21b%2F%DB4I%06%CD%C1%DDr%98%B8M%96%91&event=started&uploaded=0&port=6881
INFO 15:54:45 client_impl.net.tracker_clients.http url= http://bt2.t-ru.org/ann?compact=1&left=0&info_hash=,zP%82%7D%C3J%11%EDI%7C%FDzE%7B%EA%23%F2t&downloaded=163763027&peer_id=!-%14%B2!b%DB4I%06%CD%C1%DDr%98%B8M%96%91&event=started&uploaded=0&port=6881
DEBUG 15:54:45 client_impl.net.tracker_clients.http received response from announcer: OrderedDict([(b'interval', 3186), (b'min interval', 3186), (b'peers', b"m~\xb1b\xe8L\xe2\xb9&\x0e\xab'\xb4\xbc\xed\xbb3\x1a\xee\xfa_ IMA\xf1U\xae\xda\xeb\xc7\x08%,X\x1b2\x9c\xcd\x02#\xb7?\xee\xbb\xbb\xbc\x08\xa1\xf0A\xb0\xd4H.\xd9\x03\xb3\xb2\xe\x95Hnmj\x8f\xd9\x92\x90mV\xd7\xad\xfb\xbc\x12\xba\xc8\xd5\x97\xfcD\xab\xe8\x1cMb\xd1Z|MWu&y\xcf\x83qm)\xca\x89o\x80K\x83=\xd4\xad.\xad)\xe3\x91V\xb2\x9e\xd34\x9b5'\xb8D\x88\x84\xe90\x02=2W\xd1(\xb3\xb3YG\x1a\xe1N\xb5')6\x90\xd4\xb7\xda\xde\xb7\xa2V>|M\x97\xa6)\xefh9\xb0-\xc3\x9a\xbd\x17]T[\xd6\xf5\x9a\xd9\xc2R\xd0d\x95\xfdP")])
DEBUG 15:54:45 "Katy Perry - Witnes.." announce to "http://bt2.t-ru.org/ann" succeeded (30 peers, interval = 3186, min_interval = 3186)
DEBUG 15:54:45 "Katy Perry - Witnes.." trying to connect to 30 new peers
```

ფიგურა 12 ლოგირება

```
tsozne@tso: ~/Desktop/TSO
File Edit View Search Terminal Help
INFO 16:01:09 "ubuntu-16.04.2-desk.." progress 27.1% (805 / 2965 pieces)
DEBUG 16:01:09 "ubuntu-16.04.2-desk.." piece 191 finished
INFO 16:01:09 "ubuntu-16.04.2-desk.." progress 27.1% (806 / 2965 pieces)
DEBUG 16:01:09 "ubuntu-16.04.2-desk.." piece 1234 finished
INFO 16:01:09 "ubuntu-16.04.2-desk.." progress 27.2% (807 / 2965 pieces)
DEBUG 16:01:09 "ubuntu-16.04.2-desk.." piece 1740 finished
INFO 16:01:09 "ubuntu-16.04.2-desk.." progress 27.2% (808 / 2965 pieces)
DEBUG 16:01:10 "ubuntu-16.04.2-desk.." piece 1859 started (owned by 6 alive peers, concurrency: 6 peers)
DEBUG 16:01:10 "ubuntu-16.04.2-desk.." piece 1384 finished
INFO 16:01:10 "ubuntu-16.04.2-desk.." progress 27.2% (809 / 2965 pieces)
DEBUG 16:01:10 "ubuntu-16.04.2-desk.." piece 75 finished
INFO 16:01:10 "ubuntu-16.04.2-desk.." progress 27.3% (810 / 2965 pieces)
DEBUG 16:01:10 "ubuntu-16.04.2-desk.." piece 180 finished
INFO 16:01:10 "ubuntu-16.04.2-desk.." progress 27.3% (811 / 2965 pieces)
DEBUG 16:01:10 "ubuntu-16.04.2-desk.." piece 2116 started (owned by 6 alive peers, concurrency: 6 peers)
DEBUG 16:01:10 "ubuntu-16.04.2-desk.." piece 1775 finished
INFO 16:01:10 "ubuntu-16.04.2-desk.." progress 27.3% (812 / 2965 pieces)
DEBUG 16:01:10 "ubuntu-16.04.2-desk.." piece 2476 finished
INFO 16:01:10 "ubuntu-16.04.2-desk.." progress 27.4% (813 / 2965 pieces)
DEBUG 16:01:10 "ubuntu-16.04.2-desk.." piece 1281 started (owned by 6 alive peers, concurrency: 6 peers)
```

ფიგურა 13 ლოგირება

6.2 შეჯამება (გამოყენებული მოდულები, პროგრამის მახასიათებლები და სპეციფიკაციები)

საბოლოო შეჯამებისთვის : პროგრამა დაწერილია python3.5-ზე, გამოყენებული მოდულებია:

- aiohttp
- bencodepy
- bitarray

გრაფიკული გარსისთვის (GUI):

- PyQt5

6.2.1 TSO მახასიათებლები და სპეციფიკაციები :

მახასიათებლები:

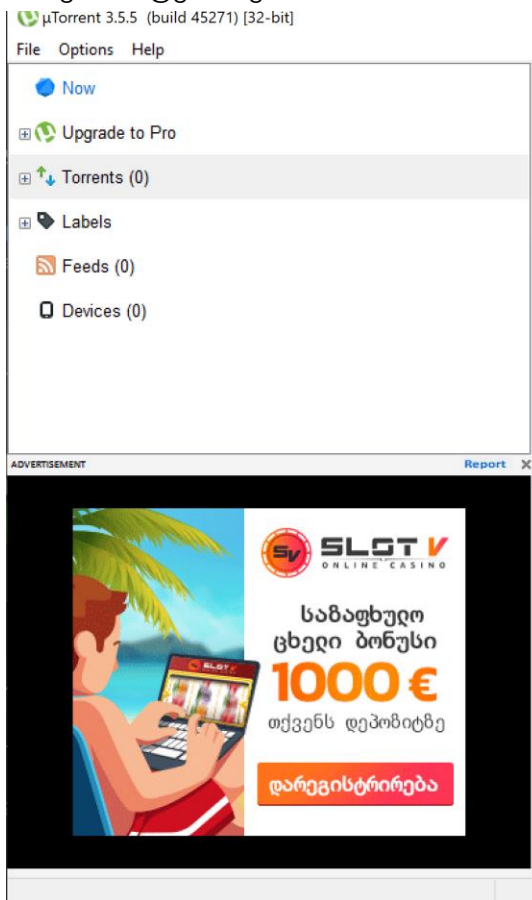
- შეგიძლიათ ჩამოწეროთ ტორენტ ფაილები და გააზიაროთ თქვენი ფაილები;
- გრაფიკული ინტერფეისი (მხარდაჭერილია ე.წ. Drag'n'Drop-ც);
- კონსოლური ინტერფეისი;
- შეგიძლიათ დააპაუზოთ გადმოწერა, ნახოთ პროგრესი, გადმოწერის სიჩქარე და მოც. სიჩქარით რა დროში დასრულდება გადმოწერა;

- ტორენტ ფაილიდან აირჩიოთ რომელი ფაილების გადმოწერა გინდათ;
- იმახსოვრებს წინა გამოყენებისას არსებულ მდგომარეობას;

სპეციფიკაციები :

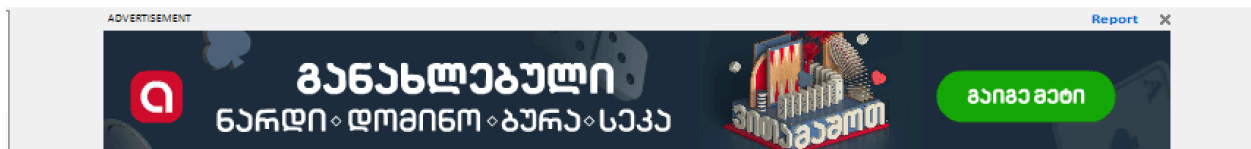
- BitTorrent პროტოკოლი (BEP 0003 რამდენიმე დამატებით);
- Multitracker Metadata Extension (BEP 0012);
- (ნაწილობრივ) UDP Tracker Protocol (BEP 0015);
- (BEP 0023);
- Type Hints(PEP 0484);
- async / await (PEP 0492);

6.3 უპირატესობები



ფიგურა 14 რეკლამა

მომხმარებელთა შორის გავრცელებულ უმრავლეს უფასო ბიტტორენტ კლიენტებს აქვთ ფასიანი ვერსიებიც, რომლებიც გარკვეული თანხის სანაცვლოდ გვთავაზობს რეკლამის გარეშე არსებულ უზრუნველყოფას, რეკლამის არსებობა შემაწუხებელს ხდის პროგრამის გამოყენებას, ასევე უმრავლესი უფასო ბიტტორენტ კლიენტი არ არის ღია კოდი. რეკლამის არსებობა და მისი გამოჩენა, ბუნებრივად ანელებს გადმოწერის სიჩქარეს, ასევე ფასიანი ვერსიის უპირატესობის საჩვენებლად კოდის დონეზეა ზოგჯერ შეზღუდული მაქსიმალური სიჩქარე. უმრავლესი ბიტტორენტ კლიენტი გარდა ამისა მომხმარებლის მონაცემებს ინახავს, მის მიერ ჩამოწერილ ფაილებს ანალიზებს და მოცემული ინფორმაციით ვაჭრობს. მეხუთე და მეექვსე ფიგურაზე წარმოდგენილია ყველაზე პოპულარული ბიტტორენტ კლიენტის



ფიგურა 15 რეკლამა

µTorrent-ის უფასო ვერსიაში არსებული რეკლამები [30].

სამაგისტრო ნაშრომის ავტორის მიერ შექმნილი კლიენტი უფასოა, ღია კოდია და არ შეიცავს არანაირ შეზღუდვას, არ აგროვებს და არ ყიდის კლიენტის ინფორმაციას. და რაც მთავრია შექმნილია უახლესი ტექნოლოგიების გამოყენებით.

7. დასკვნა

ბოლო წლებში ბიტტორენტი იქცა ყველაზე პოპულარულ ერთრანგოვანი ქსელის პროტოკოლად. ის მარტივი გამოსაყენებელია, აქვს მძლავრი შესაძლებლობები, რომელიც საშუალებას გვაძლევს ფაილები გადმოვწეროთ სწრაფად და გვთავაზობს სამართლიანობის პრინციპს ქსელის შიგნით არსებულ მომხმარებლებს შორის, განსხვავებით სხვა ერთრანგოვანი პროტოკოლებისგან. ბიტტორენტი ფაილების მიმოცვლის კარგად ჩამოყალიბებული და მძლავრი პროტოკოლია, რომელიც ეფუძნება ფართო ადაპტაციასა და ინტერნეტ მომხმარებლების გამოყენებას. ეს არ ნიშნავს რომ ის საუკეთესოადაა ოპტიმიზირებული, ყოველ წელს ის უფრო და უფრო იხვეწება და ვითარდება.

თავიდან პროტოკოლში აუცილებელი იყო ცენტრალიზებული ტრეკერი არსებობა, რათა მომხმარებლებს ეპოვათ ერთმანეთი, ანუ თავიდან ის წარმოადგენდა ჰიბრიდულ ერთრანგოვან პროტოკოლს. პროტოკოლის ახალ ვერსიაში აღარ გვაქვს ცენტრალური ტრეკერი და ამ ფუნქციას თავად მომხმარებლები ითავსებენ. ამ გაუმჯობესებამ ბიტტორენტი წმინდა ერთრანგოვან პროტოკოლად აქცია.

მოცემული პროტოკოლი თვითკონფიგურებადია. მომხმარებლები ერთმანეთს პოულობენ ცენტრალური თუ განაწილებული ტრეკერის გამოყენებით, ყოველგვარი მომხმარებლის ჩართულობის გარეშე. პროტოკოლში არსებული ალგორითმებისა და წესების გამოყენებით, მომხმარებლები ირჩევენ ვისგან გადმოიწერონ და ვის გადააწერინონ ფაილები.

ბიტტორენტს აქვს პოტენციალი, რევოლუცია მოახდინოს სამაუწყებლო მედიასა და ფაილების გავრცელების ინდუსტრიაში. შეგვიძლია წარმოვიდგინოთ მომავალი, სადაც მომხმარებლები საყვარელი გადაცემებისა თუ ტელეშოუების მხოლოდ იმ ნაწილს გადმოიწერენ რომელიც აინტერესებთ და არა მთლიანად. ის შეიძლება გამოყენებულ იქნას ფილმებისა და ტელეშოუების გავრცელების ხარჯების შესამცირებლად, რაც საშუალებას მიცემს მწარმოებლებს თავიანთი კონტენტი თითოეულ ინტერნეტ მომხმარებელს მარტივად და იაფად მიაწოდონ.

სამაგისტრო ნაშრომის შედეგად გვაქვს მულტიპლატფორმული, სრულყოფილი BitTorrent პროტოკოლთან თავსებადი, კლიენტ პროგრამული უზრუნველყოფა, რომელიც შექმნილია პროგრამირების ენა, პითონის ასინქრონული ბიბლიოთეკის

გამოყენებით - asyncio. ავტორის პროგრამა წარმოდგენილია როგორც გრაფიკული ინტერფეისით, ასევე უზრუნველყოფას გააჩნია კონსოლური ინტერფეისის მხარდაჭერაც.

8. გამოყენებული ლიტერატურა

- [1] Anthony Bellissimo, Brian N. Levine and Prashant Shenoy. Exploring the Use of BitTorrent as the Basis for a Large Trace Repository.
- [2] John Borland. BitTorrent file-swapping networks face crisis. 20.12.2004.
- [3] Clue for Windows, version 6.
- [4] Bram Cohen. Incentives Build Robustness in BitTorrent. May 22, 2003.
- [5] Bram Cohen. BitTorrent version notes.
- [6] Bram Cohen. BitTorrent Goes Trackerless: Publishing with BitTorrent gets easier!
- [7] K. Kant, R. Iyer and V. Tewari. A framework for classifying peer-to-peer technologies. Cluster Computing and Grid 2nd IEEE. 2002.
- [8] Andrew Kantor. Despite reports, Grokster decision is a win for file sharing.
- [9] David Karger, Eric Lehman, Tom Leighton, Matthew Levine, Daniel Lewin and Rina Panigrahy. Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web.
- [10] LegalTorrents.com. <http://www.legaltorrents.com/>
- [11] Petar Maymounkov and David Mazières. Kademlia: A Peer-to-peer Information System Based on the XOR Metric.
- [12] Chris Meredith. The story tit-for-tat. 1998.
- [13] Iqbal Mohamed. Understanding BitTorrent.
- [14] NextGenTel. <http://www.nextgentel.no>
- [15] Opera. Press release: Faster, more efficient downloads in Opera technical preview with BitTorrent.
- [16] Opera. Torrents. <http://www.opera.com/download/torrents/>
- [17] J.A. Pouwelse, P. Garbacki, D.H.J. Epema and H.J. Sips. A Measurement Study of the BitTorrent Peer-to-Peer File-Sharing System. Delft University of Technology.

- [18] R. Schollmeier. A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications. First International Conference on Peer-to-Peer Computing (P2P'01) IEEE. August 2001.
- [19] Supreme Court of the United States. Metro-Goldwyn-Mayer Studios Inc. v. Grokster Ltd.
- [20] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek and Hari Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. MIT Laboratory for Computer Science.
- [21] Clive Thompson. The BitTorrent Effect. WIRED Magazine, issue 13.01. January 2005.
- [22] TiVo. <http://www.tivo.com/>
- [23] Steven J. Vaughan-Nichols. There Is No Conspiracy Against BitTorrent. June 23, 2005.
- [24] Whatis.com. <http://www.whatis.com>
- [25] Brandon Wiley. Distributed Hash Tables, Part I. 01.10.2003.
- [26] Wikipedia, Distributed hash tables.
http://en.wikipedia.org/wiki/Distributed_hash_tables
- [27] Stefan Saroiu, P. Krishna Gummadi and Steven D. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. Technical Report UW-CSE-01-06-02, University of Washington, Department of Computer Science and Engineering, July 2001.
- [28] Documentation: Protocol <http://www.bittorrent.com/protocol.html>
- [29] Wikipedia, List of BitTorrent clients
http://en.wikipedia.org/wiki/List_of_BitTorrent_clients
- [30] Wikipedia, Comparison of applications supporting BitTorrent,
http://en.wikipedia.org/wiki/Comparison_of_BitTorrent_clients