Kaj prenašamo s protokolom BitTorrent?

Anton Luka Šijanec, anton@sijanec.eu

Fakulteta za računalništvo in informatiko Univerze v Ljubljani

Povzetek

V članku predstavimo metodo za učinkovito in za omrežje neinvazivno metodo prenašanja metapodatkov iz pomožnega omrežja Kademlia mainline DHT protokola BitTorrent za izmenjavo datotek. Sledi pregled/analiza z opisano metodo pridobljenih metapodatkov o datotekah na voljo v omrežju BitTorrent.

Porazdeljene razpršilne tabele (angl. distributed hash table) so razpršilne tabele, ki podatke, ponavadi so to dokumenti, strukturirani kot vrednost in njej pripadajoči ključ, hranijo distribuirano na več vozliščih, na katerih se podatki shranjujejo. V računalniških sistemih se DHT uporablja za hrambo podatkov v omrežjih P2P (angl. peer to peer), kjer se podatki vseh uporabnikov enakomerno porazdelijo med vozlišča in so tako decentralizirani in preprosto dostopni članom omrežja. Ker se podatki izmenjujejo znotraj omrežja na vozliščih, ki z izvorom in destinacijo podatkov niso povezani, jih lahko vozlišča v velikih količinah shranjujejo za potrebe statistične analize omrežja.

V raziskavi preverimo praktično zmožnost pridobivanja velike količine podatkov v omrežju BitTorrent za P2P izmenjavo datotek, nato še analiziramo pridobljene podatke. Vsaka poizvedba po seznamu imetnikov datotek vsebuje ključ podatka v DHT in se prenese preko okoli $\log_2 n$ vozlišč, kjer je n število vseh uporabnikov v omrežju. Ker vsaka poizvedba obišče tako veliko število vozlišč, lahko med poizvedbo eno drugače nepovezano vozlišče prejme veliko obstoječih ključev v omrežju, ki jih lahko uporabi za prenos metapodatkov v omrežju BitTorrent.

Osredotočili smo se le na na pridobivanje metapodatkov v omrežju BitTorrent, samih datotek, na katere se le-ti metapodatki sklicujejo in so v omrežju na voljo, ker jih ponujajo drugi računalniki, pa tako vsled tehničnih (njihove ogromne skupne velikosti) kot tudi pravnih razlogov (avtorsko zaščitena in protizakonita vsebina) nismo prenašali. Metapodatki konceptualno sicer niso shranjeni v DHT (namesto metapodatkov o datotekah so v omrežju shranjeni seznami računalnikov, od katerih si metapodatke lahko prenesemo), vendar odkrivanje njihovega obstoja omogoči DHT.

S pridobljenimi metapodatki ugotovimo, kateri odjemalci so najpopularnejši ter kakšna je razporeditev vsebine glede na tip datotek, ki je na voljo preko protokola BitTorrent.

Keywords: porazdeljena razpršilna tabela, porazdeljeni sistemi, omrežje P2P,
podatkovno rudarjenje, BitTorrent

1 Introduction

1.1 Distribucija datotek po principu P2P

Koncept P2P (angl. *peer-to-peer*) predstavlja alternativen način distribucije identičnih datotek večim odjemalcem. Namesto enega strežnika, ki iste podatke odjemalcem pošlje vsakič znova, v omrežjih P2P za distribucijo datotek vsak odjemalec podatke tako prejema kot tudi pošilja. Odjemalec prejeto vsebino tudi sam deli drugim te vsebine željanim odjemalcem, s čimer razbremeni ostale odjemalce.

Odjemalec za druge izve s pomočjo centralnega strežnika ali pa drugačnega signalizacijskega protokola. Ker se povezujejo neposredno, medsebojno poznajo svoje internetne naslove.

1.2 Protokol BitTorrent

Od 2008[1] je eden izmed popularnejših protokolov za P2P distribucijo BitTorrent, razvit že 2001[2]. Zaradi razširljive zasnove ga je moč dopolnjevati — dodajati nove funkcije. Sprva je BitTorrent temeljil na centralnih strežnikih za koordinacijo rojev, od leta 2005 pa z uvedbo protokola DHT lahko deluje povsem neodvisno.[3]

Pojem	Angleško	Razlaga
soležnik	peer	odjemni program na računalniku, za povezavo nanj potrebujemo IP naslov in vrata
roj[4]	swarm	več soležnikov, ki prenašajo datoteke nekega torrenta
torrent/ metainfo	torrent/ metainfo	datoteka z metapodatki datotek; imena, velikosti, zgoščene vrednosti in drugo
sledilnik	tracker	koordinacijski strežnik z naslovi soležnikov v rojih
koščck	piece	delček datoteke konstantne dolžine
infohash	infohash	zgoščena vrednost serializiranih podatkov pod ključem `info` v torrentu, ki unikatno opišejo ključne metapodatke o torrentu
objavi	announce	pošiljanje obvestila v DHT ali na sledilnik, da se odjemalec želi priključiti nekemu roju

Tab. 1: Slovar pojmov BitTorrenta

Za prenos je treba poznati metapodatke o obstoječih datotekah, ki so shranjeni v t. i. obliki .torrent, strojno berljivi z bencoding serializirani datoteki. Vsebujejo vsaj imena in poti datotek ter njihove zgoščene vrednosti, ime torrenta, lastnosti prenosa in velikost koščka.

V raziskavi ne iščemo soležnikov s sledilniki in ne prenašamo datotek, temveč samo prenašamo in analiziramo metapodatke.

1.3 Protokol Kademlia mainline DHT

V BitTorrent je za iskanje soležnikov v roju uporabljen DHT (angl. *distributed hash table*), ki odpravi odvisnost od sledilnika.

Pojem	Angleško	Razlaga
vozlišče[4]	node	odjemni program na računalniku
usmerjevalna tabela[4]	routing table	seznam vozlišč (IP, vrata, ID), ki ga hrani posamezno vozlišče
ID vozlišča	node ID	160-bitna ob zagonu generirana naključna vozlišču pripadajoča številka
merilo razdalje	distance metric	funkcija (XOR), ki izrazi razdaljo med vozliščema kot 160-bitno številko
koš[4]	bucket	element usmerjevalne tabele, ki glede na merilo razdalje vsebuje bližnja vozlišča

Tab. 2: Slovar pojmov Kademlie. Slovenski prevodi niso ustaljeni.

Na visokem nivoju gre za abstraktno razpršilno tabelo, shranjeno porazdeljeno na velikem omrežju vozlišč. Podpira naslednji operaciji[5]:

`get_peers` Vrne seznam soležnikov (IP naslov in vrata) za torrent, opisan z njegovim infohashom.

`announce` V seznam soležnikov za torrent, opisan z njegovim infohashom, vstavi IP naslov in vrata pošiljatelja zahteve.

V raziskavi s sodelovanjem v DHT prestrezamo obstoječe ključe v razpršilni tabeli, z njimi pridobimo soležnike, od katerih prenesemo metapodatke o torrentih za kasnejšo analizo.

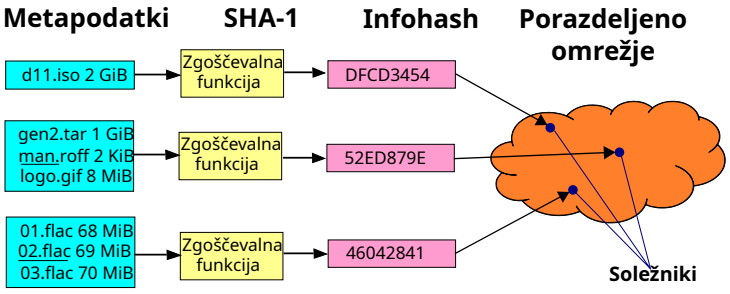

Fig. 1: Shematski prikaz DHT

2 Opis standardov

Serializacija bkodiranje (`bencoding`) V BEP-0003[6] je opisan bencoding. Z njim je serializirana večina struktur BitTorrenta in Kademlie. Bkodiranje je podobno bolj znanemu JSONu [7] — vsebuje štiri podatkovne tipe: niz, število, seznam in slovar.

Datoteka metainfo/.torrent Za distribucijo vsebine s protokolom BitTorrent ustvarimo .torrent datoteko, bkodiran slovar z metapodatki, nujnimi za prenos datotek. Za raziskavo so pomembni metapodatki pod ključem `info`:[6]

- `private`: prepoved objavljanja preko DHT, le preko sledilnikov (ti torrenti niso zajeti v raziskavi)[8]
- `name`: ime torrenta oz. datoteke za enodatotečne torrente
- `piece length`: velikost koščka — datoteke so spojene skupaj in razdeljene na enako velike koščke
- `pieces`: niz dolžine $20n$ (n je število koščkov) s SHA-1 vrednostmi koščkov[9]
- `length`: dolžina datoteke za enodatotečne torrente
- `files`: seznam datotek v večdatotečnem torrentu — vsaka datoteka je predstavljena kot slovar z `length` in `path`.

Kdor pozna infohash, lahko od soležnika prenese metainfo in posledično tudi pripadajoče datoteke. Roj najde in se vanj vključi s poizvedbo v DHT, saj je infohash ključ v tej razpršilni tabeli[10]. Infohash običajno oblikujemo v t. i. magnetno povezavo (magnet URI): `magnet:?dn=`**`ime torrenta`**`&xt=urn:btih:`**`infohash`**

Druga različica BitTorrenta ima drugačno metainfo strukturo s podobnimi podatki. Uporablja SHA-256 in namesto `pieces` uporablja `merkle hash tree`[11] za zgoščene vrednosti datotek.

Graf DHT DHT vzdržuje sezname soležnikov v roju vseh obstoječih torrentov. Vozlišča komunicirajo preko UDP in so del velikega usmerjenega grafa, vsako s $K \log_2 n$ (konstanta $K = 8$, n je število vseh vozlišč na svetu) povezavami — vpisi v usmerjevalno tabelo.

Vozlišče skrbi za urejeno usmerjevalno tabelo dosegljivih[1] vozlišč v koših; $i$ti koš hrani do K med 2^i in 2^{i-1} po merilu XOR oddaljenih vozlišč, torej je shranjenih veliko bližnjih in malo zelo oddaljenih vozlišč.[12]

Poizvedbe po grafu Sprehod po grafu med poljubnima vozliščema je torej dolg v povprečju $\log n$ (n kot prej). Roj torrenta z infohashom x je shranjen na vozliščih z ID blizu x, tedaj ima poizvedba po soležnikih/objavljanje soležnika časovno kompleksnost $O(\log n)$. Za pridobitev seznama soležnikov torrenta pošljemo bkodiran UDP paket tipa `get_peers` t[2] vozliščem iz usmerjevalne tabele, ki so blizu infohasha. Pozvana vozlišča odgovorijo s seznamom do K temu infohashu najbližjih vozlišč in seznamom soležnikov za ta torrent, če ga imajo. Novodobljenim vozliščem spet pošljemo poizvedbo `get_peers` in postopek nadaljujemo, dokler ne najdemo nekaj infohashu najbližjih vozlišč. V tista pošiljamo objave in od njih še naprej prejemamo informacije o roju.

[1] dvosmerna komunikacija zaradi NAT ni samoumevna
[2] odvisno od implementacije

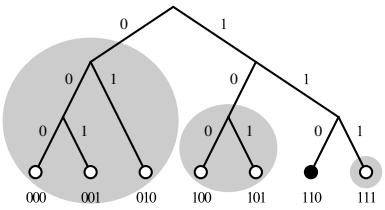

Fig. 2: Usmerjevalna tabela za vozlišče 110 (koši so osenčeni)

3 Metode

Vsaka poizvedba obišče $\log n$ vozlišč, torej vsako vozlišče v DHT prejema ogromno ključev — infohashov. V raziskavi v C spišemo program travnik, nepopolno implementacijo odjemalca BitTorrent s poudarkom na DHT. Osredotočimo se na zajem metapodatkov iz ključev, ki jih prejmemo s sodelovanjem v omrežju.

Ko vozlišče, na katerem teče travnik, prejme paket z infohashom, ga doda v seznam željenih torrentov. Neprestano posodablja roje torrentov znanih infohashov in se povezuje na soležnike iz njih, dokler mu ne uspe prenesti metapodatkov torrenta oziroma dokler ne obupa/preteče 256 sekundni TTL torrenta. Izdeluje .torrent datoteke z najdenimi metapodatki in internetnim naslovom ter ime programske opreme soležnika, od katerega je metapodatke prejel. Ne objavlja se v roj, ker ne redistribuira niti metapodatkov niti datotek.

Da program prvič začne sodelovati z omrežjem — da ga sosednja vozlišča vpišejo v svoje usmerjevalne tabele — prenese metapodatke vgrajenega torrenta `Big Buck Bunny`.

Za implementacijo spišemo knjižnico za bkodiranje in bdekodiranje, knjižnico za DHT in nekaj funkcij za prenos metapodatkov od soležnikov preko TCP.

Za obdelavo dobljenih torrent datotek uporabimo Jupyter Notebook[13] in spišemo preprosto knjižnico za razčlenjevanje .torrent metainfo datotek, ki jih generira travnik.

4 Rezultati

4.1 Zajem

Podatke smo zajemali iz različnih lokacij in v različnih časovnih obdobjih.

mesto	datum	dni	torrentov	sek./torrent
T-2, FTTH, SI	1.-2. ’23	16	47863	29
GRNET, VPS, GR	1.-2. ’23	31	412846	6
T-2, FTTH, SI	6. ’24	5	62110	7

Tab. 3: Omrežne lokacije in časovna obdobja zajemov.

Metapodatki prvega zajema opisujejo 3084321 datotek v skupni velikosti 259 TiB, metapodatki drugega zajema 17101702 datotek v velikosti 1881 TiB in metapodatki tretjega zajema 3725125 datotek v velikosti 345 TiB.

4.1.1 Primer strukture torrent datoteke z metapodatki

Spodaj je iz bencoding v JSON pretvorjena metainfo datoteka prevzetega torrenta z infohashom `696802a16728636cd72617e4cd7b64e3ca314e71`.


```
{
  "creation date": 1676645016,
  "encoding": "UTF-8",
  "info": {
    "files": [
      { "length": 5903711328,
        "path": ["John.Wick.Chapter.2.2017.2160p.4K.BluRay.x265.10bit.AAC5.1-[YTS.MX].mkv"]},
      { "length": 358, "path": [ "YTSProxies.com.txt" ] },
      { "length": 53226, "path": [ "www.YTS.MX.jpg" ] }
    ],
    "name": "John Wick Chapter 2 (2017) [2160p] [4K] [BluRay] [5.1] [YTS.MX]",
    "piece length": 4194304
  },
  "source": { "ip": "::ffff:185.242.21.95/9090", "v": "Transmission 3.00" }
}
```


4.2 Analiza

4.2.1 Odjemalci, od katerih so bili prejeti torrenti

Imenom programom odstranimo različico in jim ročno normaliziramo ime[3] ter prikažemo njihovo gostoto v populaciji.[14]

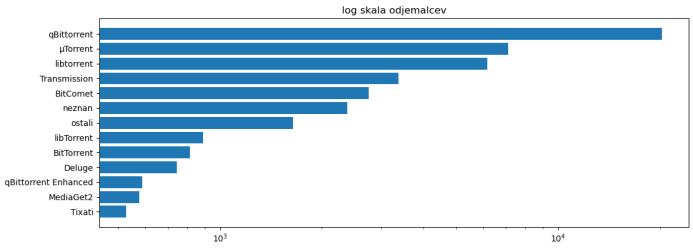

Fig. 3: Reprezentacija odjemalcev, ki predstavljajo vsaj en odstotek populacije, na logaritemski skali

4.2.2 Različice odjemalca `qBittorrent` skozi čas

Primerjava porazdelitve različic po zgornji analizi najpopularnejšega odjemalca na $\log_{10}$ skali pokaže višanje različic skozi čas. V obeh letih smo prejeli torrente od skupno 88 različnih inačic qBittorrenta. V 2023 smo največ torrentov prejeli od odjemalcev različice 4.5.0, v 2024 pa od odjemalcev različice 4.6.3.

[3] µTorrent se sicer pojavi dvakrat, enkrat ima znak mikro, enkrat pa grško črko mu. Unicode namreč ta dva znaka, ki sicer izgledata identično, hrani pod dvema različnima kodama.

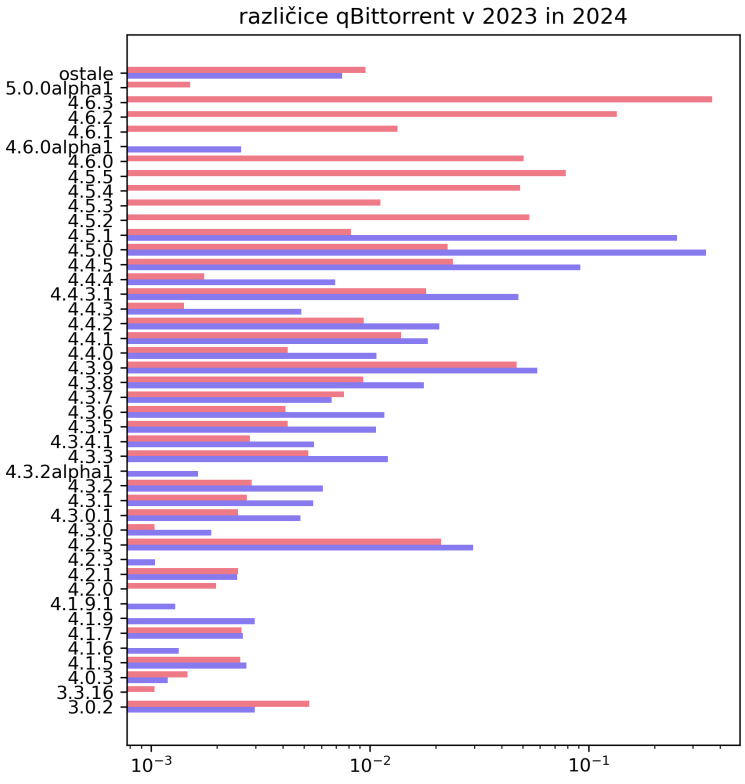

Fig. 4: Primerjava distribucij različic odjemalca `qBittorrent` med 2023 (plavo) in 2024 (roza), ki predstavljajo vsaj promil populacije (delež).

4.2.3 Geolokacija IP naslovov odjemalcev

Z uporabo podatkovne zbirke MaxMind GeoLite2[15] IP naslovom, od katerih smo prejeli torrente, določimo izvirno državo.

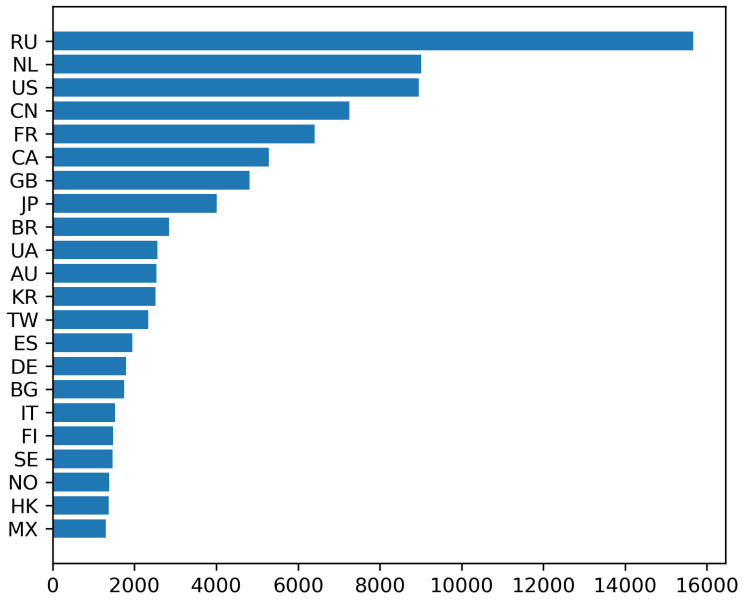

Fig. 5: Reprezentativnost držav, iz katerih smo prenesli meta-info, na linearni skali. Prikazane so le države, iz katerih izvira vsaj odstotek populacije.

4.2.4 Predstavnost ključev v prejetih slovarjih `info`

Poleg standardnih obveznih nekateri torrenti vsebujejo tudi dodatne metapodatke v slovarju info. Pogostost slednjih prikazuje spodnji grafikon.

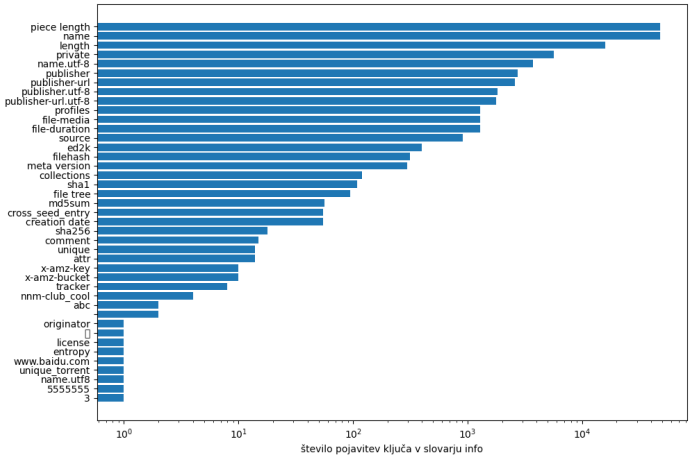

Fig. 6: Reprezentacija ključev v slovarju `info` na logaritemski skali

4.2.5 Tipi datotek, ki se prenašajo v torrentih

Iz končnice datoteke izvemo tip datoteke. Vsakemu torrentu priredimo reprezentativen tip, tisti, ki po velikosti prevladuje v torrentu. Glede na število torrentov z nekim reprezentativnim tipom kvantificiramo pogostost tega datotečnega tipa za tipe, ki zavzemajo vsaj promil populacije.

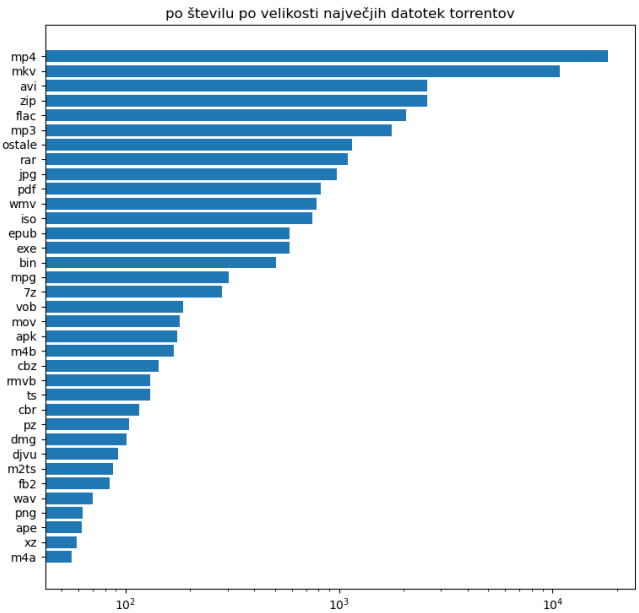

Fig. 7: Reprezentativni tipi torrentov, ki predstavljajo vsaj en promil populacije, na logaritemski skali

Razvidno je, da je večina torrentov namenjena prenosu videovsebin, zvočnih datotek in stisnjenih arhivov.

Če bi za določilo pojavnosti tipa uporabili število datotek, bi prevladovali tipi vsebin, ki so ponavadi preneseni kot kopica datotek, denimo slike (diagram v prilogi na sliki 8), če pa bi za določilo pojavnosti tipa uporabili velikost datotek tega tipa, pa bi prevladovali tisti tipi, ki zasedajo več prostora. V tem primeru bi npr. videovsebine zaradi svoje velikosti občutno presegale digitalne knjige (diagram v prilogi na sliki 9).

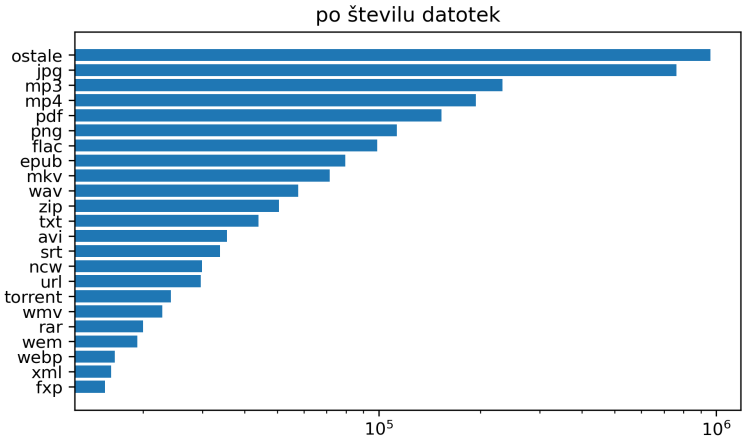

Fig. 8: Pojavnost tipa kot število datotek

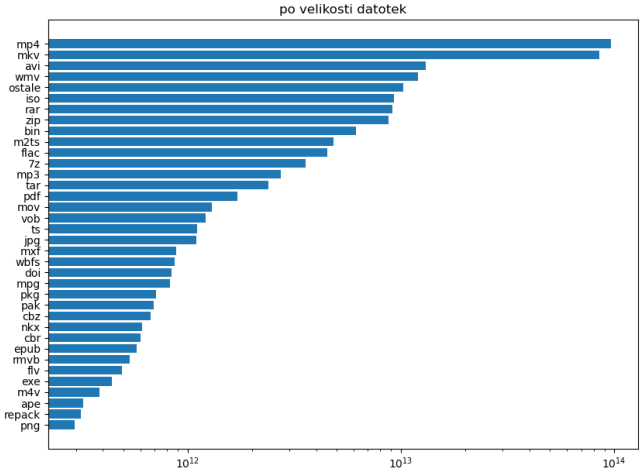

Fig. 9: Pojavnost tipa kot velikost datotek (tipi, ki zavzamejo vsaj odstotek populacije)

4.3 Diskusija

Statistična kvaliteta vzorca Zaradi lastnosti uniformne porazdelitve zgoščevalne funkcije[16, 5.2] mesto infohasha na intervalu vseh možnih infohashov ni odvisno od metapodatkov. Kot posledico načina vzorčenja z DHT pričakujemo, da je porazdelitev infohashov prejetih torrentov po celotnem sprektru števil z intervala $\left[0, 2^{160} - 1\right]$ gostejša okoli IDja vozlišča, ki ga je med prenašanjem imelo naše iskalno vozlišče. IDje smo tekom raziskave izbrali naključno na vsaki merilni lokaciji in jih med meritvijo tudi nekajkrat zamenjali. Kljub temu je vsled nepovezanosti vsebine in infohasha vzorec še vedno statistično reprezentativen. Zajem ne more biti pristranski glede na metapodatke, ker nikjer v procesu zajema ne obravnavamo torrentov glede na metainfo, temveč le glede na infohash.

Težava z zajemom podatkov Vsled majhne velikosti UDP paketov DHT glavno ozko grlo pri zajemu predstavlja število paketov, ki jih mrežna oprema lahko posreduje v sekundi. Domača optična povezava dopušča do okoli 2000 paketov na sekundo na naključno porazdeljene IP naslove, odjemno mesto na VPS pa je imelo to omejitev veliko višjo, zato smo tam v istem časovnem intervalu shranili veliko več torrent datotek.

Etičnost in legitimnost rudarjenja podatkov Čeprav gre za izrazito osebne podatke, se morajo uporabniki BitTorrent omrežja zavedati, da so njihovi prenosi *a priori* javni, tudi če jih nihče aktivno ne zajema. Nekateri BitTorrent odjemalci uporabnike ob prvem zagonu o tem obvestijo.

5 Priloge

Izvorna koda programa travnik in ipynb datotek za analizo podatkov je na voljo na http://ni.šijanec.eu./sijanec/travnik.

Korpus zajetih metapodatkov je na voljo na rsync://b.sjanec.eu./travnik.

Literatura

[1] D. Harrison. (2008) Index of bittorrent enhancement proposals. [Online]. Available: http://www.bittorrent.org/beps/bep_0000.html

[2] B. Cohen. (2001) Bittorrent - a new p2p app. Internet Archive. [Online]. Available: http://finance.groups.yahoo.com/group/decentralization/message/3160

[3] B. Jones. (2015) Bittorrent's dht turns 10 years old. [Online]. Available: https://torrentfreak.com/bittorrents-dht-turns-10-years-old-150607/

[4] M. Gams *et al.* (2008) Dis slovarček. [Online]. Available: https://dis-slovarcek.ijs.si/

[5] A. Loewenstern and A. Norberg. (2020) Dht protocol. [Online]. Available: https://www.bittorrent.org/beps/bep_0005.html

[6] B. Choen. (2017) The bittorrent protocol specification. [Online]. Available: https://www.bittorrent.org/beps/bep_0003.html

[7] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, and D. Vrgoč, "Foundations of json schema," in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 263–273.

[8] D. Harrison. (2008) Private torrents. [Online]. Available: https://www.bittorrent.org/beps/bep_0027.html

[9] D. Eastlake and P. Jones, "US Secure Hash Algorithm 1 (SHA1)," RFC 3174, 9 2001.

[10] G. Hazel and A. Norberg. (2017) Extension for peers to send metadata files. [Online]. Available: https://www.bittorrent.org/beps/bep_0009.html

[11] B. Cohen. (2017) The bittorrent protocol specification v2. [Online]. Available: https://www.bittorrent.org/beps/bep_0052.html

[12] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *Peer-to-Peer Systems: First InternationalWorkshop, IPTPS 2002 Cambridge, MA, USA, March 7–8, 2002 Revised Papers*. Springer, 2002, pp. 53–65.

[13] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, "Jupyter notebooks – a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Schmidt, Eds. IOS Press, 2016, pp. 87 – 90.

[14] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[15] MaxMind. (2024) Geolite2 free geolocation data. [Online]. Available: https://dev.maxmind.com/geoip/geolite2-free-geolocation-data

[16] D. E. E. 3rd, S. Crocker, and J. I. Schiller, "Randomness Requirements for Security," RFC 4086, Jun. 2005. [Online]. Available: https://www.rfc-editor.org/info/rfc4086

Viri slik

- Slika 2: Limaner: nespremenjena, izvorna pod CC BY-SA

Dovoljenje