თბილისის სახელმწიფო უნივერსიტეტის ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტი

ინტერნეტის ნეიტრალიტეტი

ხელმძღვანელი:
ნანა ოდიშელიძე
მომხსენებლები:
გიორგი ბაღდავაძე
სანდრო ღვინჯილია

2018

სარჩევი

აბსტრაქტი

დღესდღეობით ინტერნეტი ჩვენი ყოველდღიური ცხოვრების დიდ ნაწილს წარმოადგენს, მისი მეშვეობით, ყოველდღიურად ვასრულებთ მრავალ პროცესს, რომლებიც სხვადასხვა სერვისების სახით არის წარმოდგენილი და გვეხმარება როგორც პროდუქტიულობის ამაღლებაში ან სხვადასხვა სამუშოს შესრულებაში, ასევე არის განტვირთვის, გართობისა და რელაქსაციის საშუალებაც. განურჩევლად იმისა თუ რომელ სერვისს ვიყენებთ, ინტერნეტ სერვისის პროვაიდერებს ვუხდით ფიქსირებულ თანხას, რომელიც წვდომას გვაძლევს მთელს ინტერნეტზე, თუმცა წარმოიდგინეთ ინტერნეტი,

რომელშიც სხვადასხვა სერვისების გამოსაყენებლად მოგვიწევდა დამატებითი საფასურის გადახდა, მოგვიწევდა არა იმ სერვისების გამოყენება, რომლებიც მოგვწონს, არამედ რომელზეც ხელი მიგვიწვდება. ეს რეალობისაგან არც ისე შორია, და შეიძლება ახლო მომავალში ინტერნეტი მხოლოდ ესეთი, არა-ნეიტრალური სახით ვიხილოთ. როგორ მივედით ამ მდგომარეობამდე და როგორ ავიცილოთ თავიდან მომავალი გაუარესება, აღწერილია რეფერატში.

Abstract

Nowadays, the Internet is a great part of our everyday life, and we carry on a daily basis a number of processes that are presented as different services and help us to enhance productivity or perform various works, also they provide a way of relaxation and entertainment. Regardless of which service we use, Internet service providers charge us a fixed amount costs to access the internet, but imagine the Internet in which we would have to pay an additional fee to use different services. In that case we would have to use the services that we can instead of the services we like. It's not too far from reality, and in the nearest future we might see the Internet only in a non-neutral way. How did we come to this situation and how to avoid future deterioration is described in this paper.

გასაღები სიტყვები

დეცენტრალიზებული ინტერნეტი, ინტერნეტის ნეიტრალიტეტი, ინტერნეტის/ვების მომავალი, მიუკერძოებელი ინტერნეტი.

Keywords

Decentralized Internet, Internet neutrality, Future of internet / web, Impartial internet.

შესავალი

რა არის ინტერნეტის ნეიტრალიტეტი?

ნეიტრალური (მიუკერძოებელი) ინტერნეტი (ან მეორენაირად, ღია ინტერნეტი) არის ქსელის პრინციპი, რომლის მიხედვითაც ინტერნეტ-პროვაიდერებმა და მთავრობებმა, რომლებიც ინტერნეტის უმეტესობს ნაწილს აკონტროლებენ და მასზე რეგულაციებს აწესებენ, უნდა:

- უზრუნველყონ ინტერნეტში არსებული ყველა მონაცემის თანასწორობა,
- არ იყვნენ დისკრიმინაციული ან გადაახდევინონ სხვადასხვა თანხა მომხმარებელს, შინაარსის, ვებსაიტის, პლატფორმის, აპლიკაციის ან კომუნიკაციის მეთოდის მიხედვით.

მაგალითად, ამ პრინციპების თანახმად, ინტერნეტ-პროვაიდერებს არ შეუძლიათ განზრახ დაბლოკონ, შეაჩერონ ან დააკისროს გადასახადი რაიმე კონკრეტული ტიპის ვებ საიტებს და ონლაინ კონტენტს.

ეს ტერმინი, პირველად, 2003 წელს კოლუმბიის უნივერსიტეტის მედიის სამართლის პროფესორმა, ტიმ ვუმ გამოიყენა.

ქსელური ნეიტრალიტეტის პრინციპების დარღვევის ფართოდ გავრცელებული მაგალითიებია:

- ინტერნეტ-პროვაიდერის, Comcast-ის მიერ peer-to-peer (P2P) პროტოკოლებით, ფაილების ასატვირთი აპლიკაციების საიდუმლო შენელება (მაგალითად BitTorrent), ყალბი (forged) პაკეტის გამოყენებით.
- The Madison River Comunications კომპანია 2004 წელს FCC-ს (ამერიკის ფედერაციული კომუნიკაციების კომისია) მიერ 15,000 აშშ დოლარით დაჯარიმდა, რადგან მათ შეწყვიტეს მომხმარებელთა წვდომა კონკურენტი კომპანიის, Vonage-ს სერვისებზე.
- AT&T-ც კი იყო შემჩნეული ინტერნეტის ნეიტრალიტეტის პრინციპის დარღვევაში, როდესაც მათ FaceTime-ს დაადეს შეზღუდვა, იმისათვის, რომ თავისი პრემიუმ პაკეტის (რომელიც არ ანელებდა FaceTime-ს) გაყიდვები გაეზარდათ.
- 2017 წლის ივლისში Verizon Wireless-ს ბრალი დასდეს მომხმარებლების ტრაფიკის შენელებაში, მას შემდეგ, რაც Netflix-ზე და Youtube-ზე ვიდეოები ნელა იტვირთებოდა, თუმცა Verizon-მა იმით იმართლა თავი, რომ "ქსელის ტესტირებას" ახორციელებდა.

2017 წლის აპრილში შეერთებულ შტატებში ქსელის ნეიტრალიტეტის პრინციპებზე კომპრომისის განხილვა დაიწყო. 2017 წლის 16 მაისს, ინტერნეტის ღია წესების დაცვაზე 2015 წლიდან არსებული კანონების გაუქმებაზე შედგა მოლაპარაკებები.

ძლიერი საზოგადოებრივი აზრის დამსახურებით (რომელიც ინტერნეტის ნეიტრალიტეტს ემხრობა) მრავალი ევროპული ქვეყნის მთავრობამ ინტერნეტ სერვისები აღიარა როგორც კომუნალური (როგორც ელექტროენერგია, გაზი და წყალი), რაც უწესებს პროვაიდერებს ლიმიტებს და არ აძლევს მათ უფლებას გაატარონ ღია ინტერნეტის საწინააღმდეგო რეგულაციები.

ინტერნეტ ნეიტრალიტეტის დარღვევის ტიპები

არსებობს ოთხი ძირითადი გზა, რომლებიც შეიძლება გამოყენებული იქნას ინტერნეტ ნეიტრალიტეტის დასარღვევად, ესენია:

- **პროტოკოლით დისკრიმინაცია** არის გარკვეული ინფორმაციის დაბლოკვა (ან ხელშეწყობა) იმის მიხედვით თუ რომელ საკომუნიკაციო პროტოკოლს იყენებენ კომპიუტერები ინფორმაციის გაცვლისათვის. მაგალითად, როგორც გემოთ იყო ნახსენები, P2P ფაილების გაცვლის შენელება არის ამის მაგალითი.

- **IP მისამართით დისკრიმინაცია** - 2000-იანი წლების დასაწყისში, კომპანია NetScreen-მა, მავნე პროგრამების (Malware) გასაფილტრი firewall-ების ქსელი შექმნა, რომელიც პაკეტების ღრმა ინსპექციას აკეთებდა (deep packet inspection). ამის მეშვეობით, რეალურ დროში იყო შესაძლებელი სხვადასხვა ტიპის მონაცემების გარჩევა და მათი წარმომავლობის დადგენა, რაც შემდგომში დაცვის მექანიზმიდან ინტერნეტ ცენზურაში გადაიზარდა. ამის მეშვეობით მივიღეთ ე.წ. დასპონსორებული (უფასო) ინფორმაციის კონცეპტი (zero-rated, toll-free, sponsored data), როდესაც ინტერნეტ სერვისის პროვაიდერები, გარკვეული კომპანიების სხვადასხვა ვებ-სერვისების გამოყენებისას გადასახადს არ გვაკისრებენ. ამის მაგალითებია Facebook Zero და Google Free Zone, რომლებიც შესაბამისად, ფეისბუქისა და გუგლის გარკვეული სერვისების უფასოდ გამოყენებას გვთავაზობენ. ეს პრაქტიკა განსაკუთრებით ხშირი განვითარებად ქვეყნებშია. მოკლედ რომ ვთქვათ, ISP-ების მიერ, ზოგი კომპანიის ტრაფიკის უფასოდ გაცემა, ზოგის კი პირიქით, მომატებული ფასით, ეწინააღმდეგება ინტერნეტ ნეიტრალიტეტის პრინციპებს.

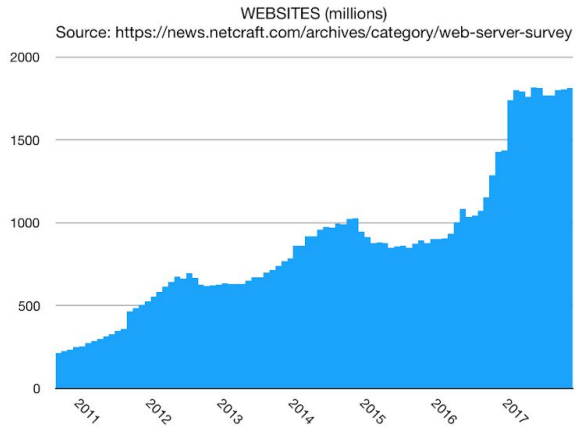
- **კერძო ქსელების ხელშეწყობა** არის წინა ორ დარღვევაზე თავის არიდების გზა, რომელიც ჯერ არ არის რეგულირებული. ის საშუალებას აძლევს ინტერნეტ-პროვაიდერებს შექმნან საკუთარი ქსელებში, სადაც გამოიყენებენ საკუთარ პროტოკოლებს, და უპირატესობას მიანიჭებენ მხოლოდ მათ, შესაბამისად მხოლოდ ის სერვისები, რომლებიც მათ პროტოკოლს გამოიყენებს იქნება ადვილად ხელმისაწვდომი.

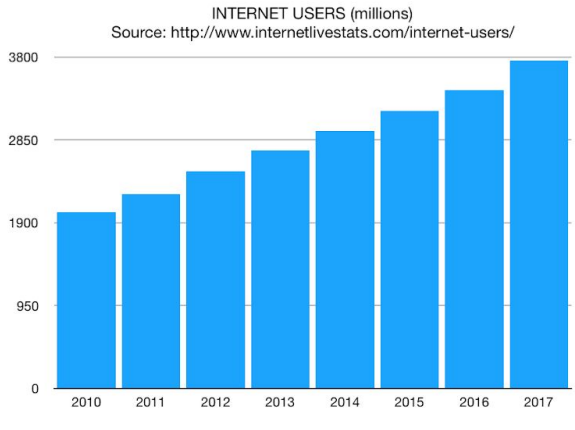
- **გარკვეული ვებ-გვერდების ჩატვირთვის აჩქარება/შენელება** ზემოთ აღნიშნულ დარღვევებთან შედარებით, ბევრად მარტივია, მაგალითად, ISP-ებს შეუძლიათ მათთვის ხელსაყრელი ვებ-სერვისების ჩატვირთვის მიზერიული დროებით შენელება, რაც გამოსაკვლევად და აღმოსაჩენად რთულია, თუმცა დიდ გავლენას ახდენს მომხმარებელზე:

კვლევები აჩვენებს რომ გვერდის ჩატვირთვის 1 წამით დაგვიანებაც კი განაპირობებს: 11%-ით ნაკლები გვერდების ნახვა და სამომხმარებლო კმაყოფილების 16%-ით შემცირება. ამასთან გამოკითხვის თანახმად, ინტერნეტ მაღაზიების მომხმარებლები, გვერდის ჩატვირთვას მომენალურად ელიან, სხვა შემთხვევაში ურჩევნიათ გახსნან სხვა მაღაზია, დალოდების მაგივრად.

არსებული მდგომარეობა

2014 წლამდე ბევრი ადამიანი სარგებლობდა გუგლით (Google), ფეისბუქით (Facebook), ამაზონით (Amazon). დღესდღეობით ისევ უამრავი ადამიანი სარგებლობს ტექნოლოგიური გიგანტების სერვისით (შესაბამისად, GOOG, FB, AMZN). დიდწილად არაფერი შეცვლილა, მეტიც, მომხმარებლისთვის შეთავაზებული ინტერფეისი და მახასიათებლები თითქმის იგივე დარჩა. თუმცა, ეს აისბერგის მხოლოდ ხილული მხარეა და სინამდვილეში ინტერნეტ სამყაროს დინამიკა და სტრუქტურა რადიკალურად შეიცვალა. ქსელში, ამ ძირფესვიანი რეფორმის ავტორები კი ზემოთ ჩამოთვლილ სამი გიგანტი კომპანია არიან. მხოლოდ გუგლს და ფეისბუქს აქვთ პირდაპირი გავლენა ინტერნეტ ტრაფიკის 70%-ზე.





(წყარო : https://news.netcraft.com/archives/category/web-server-survey and http://www.internetlivestats.com/internet-users/)

მობილური ტელეფონებიდან ინტერნეტში წვდომის ტრაფიკს უჭირავს ყველაზე დიდი წილი მსოფლიოს ქსელის ტრაფიკებში. მხოლოდ ლათინურ ამერიკაში 2016 წელს გუგლის და ფეისბუქის სერვისების მოხმარების წილი მობილურ ტელეფონებში შეადგენდა 60%-ს. 2017 წელს კი ეს ნიშნული 70%-ამდე ავიდა. შეგვიძლია ვთქვათ რომ ამ რეგიონში მობილური მოწყობილობა ძირითადად გამოიყენება გუგლისა და ფეისბუქის სერვისის წვდომისთვის.

Upstream		Downstream		Aggregate	
Facebook	30.49%	YouTube	26.09%	YouTube	23.91%
WhatsApp	15.76%	Facebook	22.92%	Facebook	23.55%
Google Cloud	11.96%	HTTP - OTHER	8.00%	HTTP - OTHER	7.70%
YouTube	6.18%	WhatsApp	7.98%	WhatsApp	7.43%
SSL - OTHER	5.94%	Instagram	4.91%	Google Market	5.85%
HTTP - OTHER	5.26%	Google Market	4.64%	Instagram	4.65%
Instagram	2.55%	MPEG - OTHER	4.46%	Google Cloud	4.41%
Google Market	1.57%	Google	3.50%	MPEG - OTHER	4.05%
MPEG - OTHER	0.94%	SSL - OTHER	2.95%	SSL - OTHER	3.27%
Snapchat	0.79%	Snapchat	1.02%	Snapchat	0.98%
	81.44%		86.28%		85.51%

sandvine

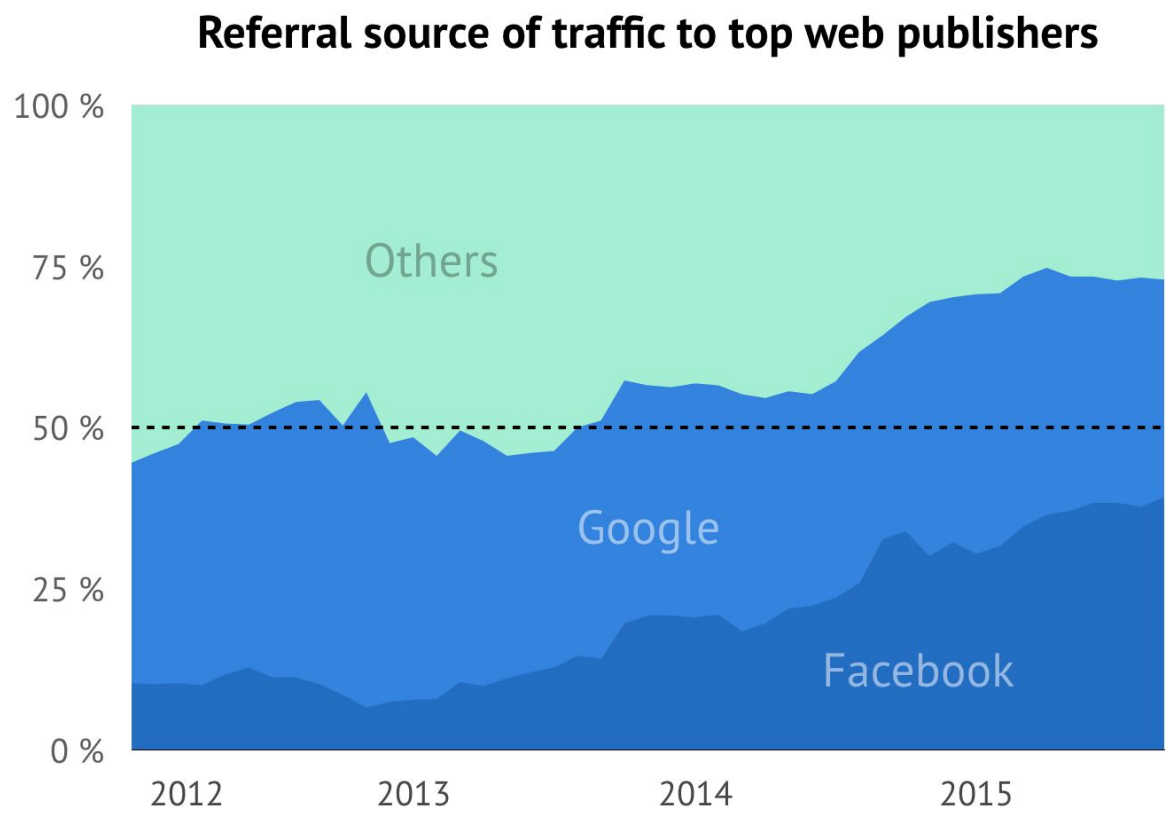
Table 4 - Top 10 Peak Period Applications - Latin America, Mobile Access

(წყარო:https://www.sandvine.com/resources/global-internet-phenomena/2016/north-america-and-latin-america.html)

მეორე მაგალითი იმისა, რომ გუგლი და ფეისბუქი დომინირებენ ვებ სივრცეში არის მედია საიტები. ამერიკაში ტოპ 10 საიტიდან 6 მედია ტიპისაა. ბრაზილიაშიც ანალოგიური მაჩვენებელია. ინგლისში კი ათიდან ხუთი ვებ-საიტი მედიას ეკუთვნის.

საიდან იღებენ მედია საიტები მათ ტრაფიკს ?

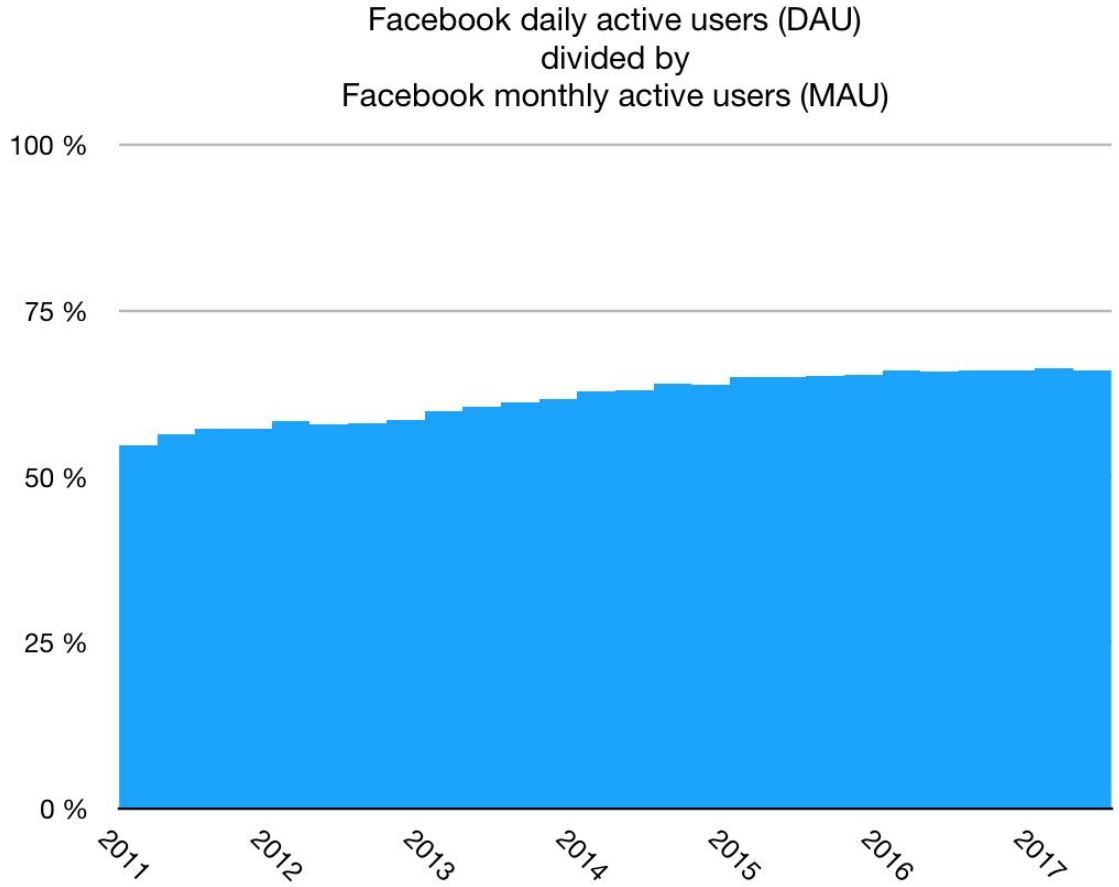
2014 წლამდე საძიებო სისტემის ოპტიმიზაცია(SEO) იყო საერთო პრაქტიკა ვებ დეველოპერებს შორის. რათა გაეუმჯობესებინათ მათი საიტების რანგი გუგლის საძიებო სისტემაში. რადგან გუგლი აღრიცხავდა ტრაფიკების 35%-ს. ხოლო 50% ზე მეტი მოდიოდა მსოფლიოს სხვადასხვა ადგილიდან. SEO - იყო მნიშვნელოვანი ხოლო ფეისბუქთან სიახლოვე დამატებით უპირატესობას წარმოადგენდა. 3 წლის შემდეგ ფეისბუქის წილი იმავე კონფიგურაციაში გაიზარდა 45 პროცენტამდე. 2017 წელს კი მედია საიტების დღიური ნახვების წილი მკვეთრად არის დამოკიდებული ფეისბუქზე და გუგლზე.



Source: https://blog.parse.ly/post/2855/facebook-continues-to-beat-google-in-sending-traffic-to-top-publishers/

ოფიციალური სტატისტიკით ფეისბუქის დომინირება ვებ სივრცეში მკვეთრად გაიზარდა. გუგლის საძიებო სისტემა კი თითქმის არ შეცვლილა. ამიტომ მან გადაწვიტა სტრატეგია

შეეცვალა და დაიწყო სოციულური მარკეტის მიმართულებით განვითარება. პირველად გაუშვა Google Wave , შემდეგ Google Buzz, Alphabet, Orkut, და Google+. ამის გარდა მან შეიძინა ჯამურად 18 სოციუალურ მედიაზე ორიენტირებული კომპანია. ფეისბუქი ამ დროს ცდილობდა შემოსულიყო საძიებო სისტემის მარკეტზე და შეიძინა Bing ის წილი კომპანია მაიქროსოფტისგან. 2014 წლის განმავლობაში ფეისბუქმა გადაწყვიტა მთელი მისი რესურსები მიემართა სოციალური ქსელების მიმართულებით. თებერვალში მან შეიძინა WhatsApp 11-ჯერ უფრო ძვირად ვიდრე გუგლმა Youtube. დეკემბერში მან შეწყვიტა თანამშრომლობა მაიქროსოფთთან. წლის ბოლოს კი მისი გავლენა ვებ სივრცეში მკვეთრად გაიზარდა. პროდუქტით როგორებიცაა: Facebook , WhatsApp, Instagram, Messenger. ფეისბუქი გახდა სოციალური გიგანტი.



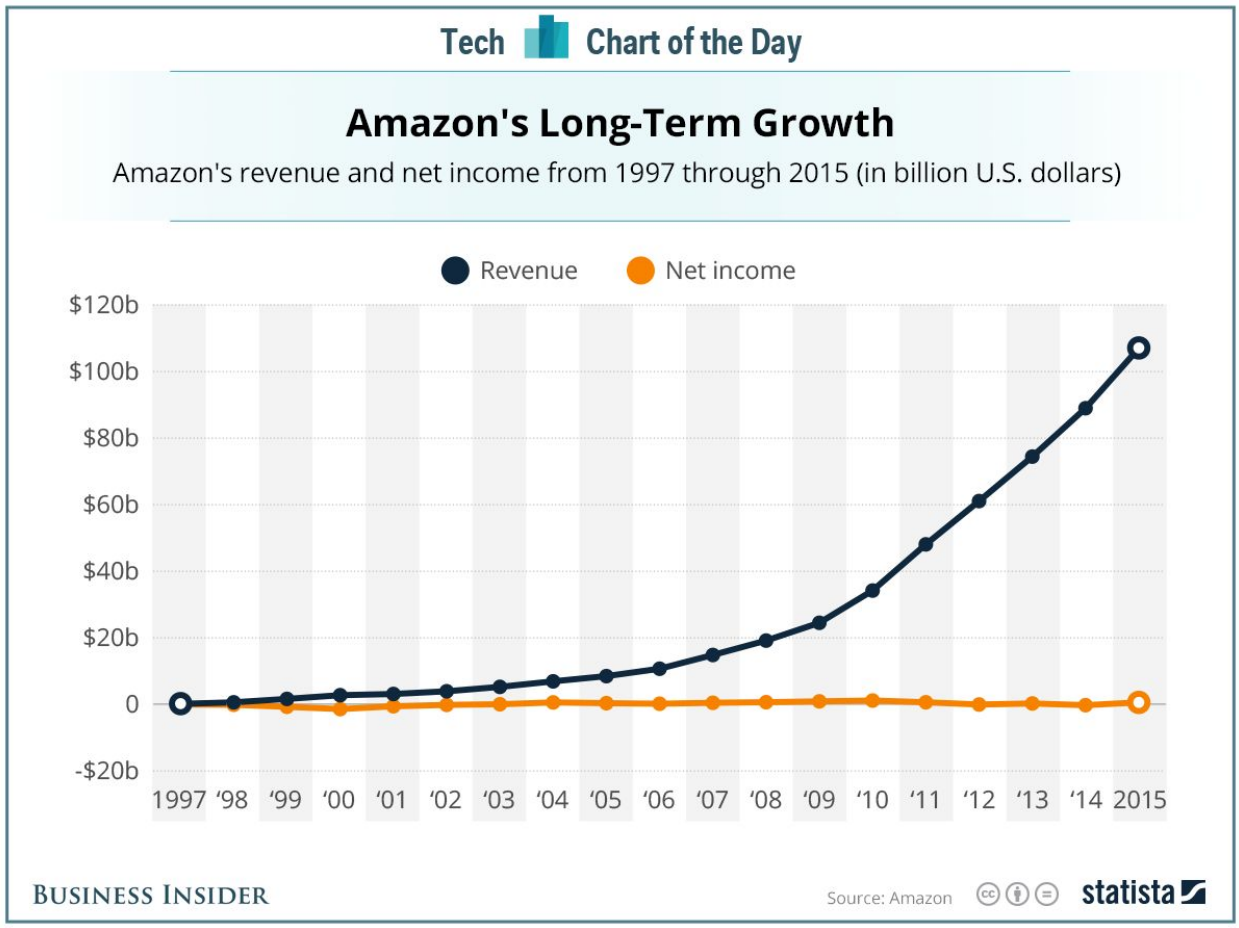
ანალოგიურად გუგლმა განიცადა რეორგანიზაცია 2014 წელს და გადაწყვიტა , რომ მთელი კონცენტრაცია გადაეტანა ხელოვნური ინტელექტის განვითარებისკენ. 2014 წლის იანვარში მან იყიდა კომპანია DeepMind, და ამავე წლის სექტემბერში მან დახურა Orkut-ი, რომელიც

სოციალური მარკეტზე ცდილობდა ფეისბუქისთვის კონკურენცია გაეწია. მან განაცხადა რომ არ ხედავს მომავალს უბრალო საძიებო სისტემაში და გადაწყვიტა ხელოვნური ინტელექტის მეშვეობით მისი შესაძლებლობები გაეუმჯობესებინა და მომხმარებლისთვის უფრო მოხერხებული და გამოსადეგი გაეხადა. “From Search to Suggest” განაცხადა Eric Schmidt-მა და დააანონსა სრული რეკონსტრუქცია. ამჟამად გუგლის ვებ სივრცეზე გავლენა საგრძნობლად ჩამორჩება ფეისბუქისას, მაგრამ, მათი გადაწყვეტილებებით ფაქტია, რომ კომპანია პროგრესირებს.

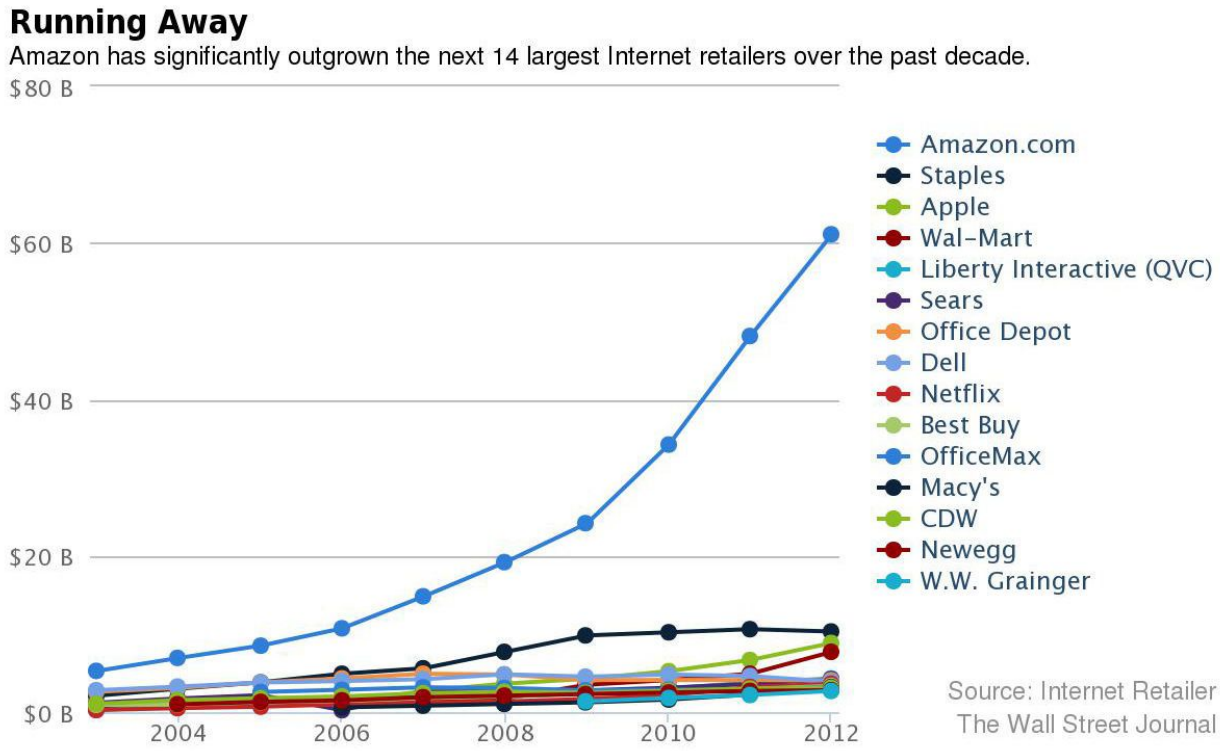
2014 წლიდან დღემდე გუგლი და ფეისბუქი მუდმივად კონკურირებდნენ. შედეგად მათ განიცადეს ევოლუცია და მიანიჭეს სხვადასხვა მიმართულებებს პრიორიტეტები. დღესდღეობით მათ ბოლომდე ათვისებული აქვთ ეს მიმართულებები და სხვადასხვა სფეროებში დომინირებენ, შესაბამისად რჩება უფრო და უფრო ნაკლები ალტერნატივა.

რაც შეეხება კომერციულ სფეროს აქ შემოდის კიდევ ერთი გიგანტი კომპანია ამაზონი.

ამაზონის წლიური შემოსავალი და მოგება გამოსახულია გრაფიკზე, რომლიდანაც ჩანს , რომ ის არ არის ორიენტირებული მოგებაზე. მისი მთავარი მიზანია შემოვიდეს ბაზარზე რაც შეიძლება ღრმად და ჩაძიროს ყველა მისი კონკურენტი კომპანია.



შემდეგ გრაფიკში კარგად ჩანს ამაზონისა და სხვადასვა ვებ სივრცეში ორიენტირებული კომპანიების კონკურირების შედეგები.



რა იყო ინტერნეტი და რა არის ის დღეს

ვების ნამდვილი იდეა რისთვისაც ის შექმნა ტიმ ბერნერს-ლიმ იყო რომ, ადამიანებს შეძლებოდათ განუსაზღვრელი ინფორმაციის გამოქვეყნება და შემდეგ მისი გამოყენება. ვები უნდა ყოფილიყო ყველასთვის თანასწორი და არა დამოკიდებული ადამიანების კონკრეტულ ჯგუფზე. ბერნერს-ლის აზრით ვები ის აღარაა რაც მან შექმნა.

1990 წლიდან 2010 წლამდე ვები, რომელსაც ჩვენ ვიყენებდით არ იყო სრულყოფილი მაგრამ ერთგულად ემსახურებოდა თავის ყველაზე მნიშვნელოვან მიზანს. ვების მრავალფეროვნებამ მრავალ კერძო ბიზნესს მოუტანა წარმატება. განვითარდა კომუნიკაცია და ურთიერთბა ხალხს შორის ფორუმების სახით. პერსონალური საიტები იჰოსტებოდა სხვადასხა სერვერებზე.

2014 წლიდან კი დაიკარგა ვების სტრუქტურის ხიბლი და ეკონომიკური დამოუკიდებლობა. რთულია კონკურენცია გაუწიო ამაზონს და გუგლს Cloud სერვისებს, რომლებიც ჰოსტავენ მსოფლიოში საიტების უმრავლესობას. ასევე ნებისმიერი ვებ საიტის პოპულარობა და რეიტინგი დამოკიდებულია გუგლის ძებნის ძრავის ალგორითმზე.

რა დაემართება ვებს მომავალში

ომი ინტერნეტის ნეიტრალიტეტის მოსაპოვებლად ამერიკაში 2014 წელს გამარჯვებით დასრულდა, მაგრამ მეორე ბრძოლა იგივე მიზნისთვის შეიძლება ითქვას წაგებულია. ინტერნეტ სერვის პორვაიდერები (ISP) მომავალში გააკონტროლებენ რომელმა ტრაფიქკებმა უნდა მიაღწიონ მომხმარებლებამდე და რომლებმა არა. რა თქმა უნდა გუგლი-ამაზონი-ფეისბუქი იქნება ყველაზე ხშირი ტრაფიკების ადრესატი ვინაიდან ისინი ისარგებლებენ ყველაზე დიდი პოპულარობით. ბაზრის მოთხოვნის გამო, პროვაიდერები უბრუნველყოფენ შედარებით იაფ ტარიფებს გუგლში, ამაზონისა, ფეისბუქითან კავშირისთვის. ამავდროულად მომხმარებლებს შეთავაზებენ უფრო ძვირ ტარიფს მთლიან ვებში შეღწევისთვის. **პორტუგალიაში ეს უკვე რეალობაა.** ეს რეალობა ეკონომიკურად გააძლიერებს ამ სამ გიგანტს და მივიღებთ შედეგს სადაც მცირე ბიზნესებს არ ექნებათ სურვილი შექმნან პერსონალური საიტები ტრაფიკის სიმცირის გამო და გადაწყვეტენ თავიან პროდუქტს რეკლამირება გაუწიონ ფეისბუქზე. მცირე კომერციულ საიტებს ან ამაზონი იყიდის ან გაკოტრდებიან რადგან ვერ გაუწევენ კონკურენციას. რადგან მოსახლეობის ნაწილს არ ექნება წვდომა მთლიან ინტერნეტთან, გუგლს ექნება სხვადასხა წახალისებები რაზეც კიდევ უფრო მეტი ადამიანი იქნება დამოკიდებული.

პრობლემის აღწერა: როდესაც ვიყენებთ ინტერნეტს და შევდივართ სხვადასხვა ვებ-გვერდებზე, ჩვენ გაუცნობიერებლად ვიყენებთ მრავალ ცენტრალიზებულ რესურსს, როგორებიცაა: რეზერვირებული ფოტოები (Stock photos), ვიდეოების ჰოსთინგები (ვებგვერდები ვიდეოებს ინახავან ისეთი სერვისებით როგორებიცაა: Youtube, Vimeo, ა.შ.), ხშირად გამოყენებული ბიბლიოთეკების სკრიპტები, რომლებიც კონტენტის მოწოდების ქსელების (CDN - Content Delivery Network) საშუალებით იტვირთება ბროუზერში და ა.შ.

ეს სერვისები, რომლებიც ზემოთ ხსენებულ რესურსებს გვაწვდიან, ძირითად რამდენიმე კომპანიის ხელშია თავმოყრილი, მაგალითად: JQuery, AngularJS, Angular, Polymer ამ ბიბლიოთეკებს ძირითადად Google-ს CDN-ებით ვიწერთ, React/Redux - Facebook-ის, ასევე საკუთარი CDN-ები აქვთ Microsoft-ს, Amazon-ს. რუსული რეგიონისათვის Yandex CDN არის გავრცელებული ხოლო ჩინეთში Baidu. აქედან გამომდინარე, იმ შემთხვევაში თუ თვისუფალი ინტერნეტის პრინციპები დაირღვევა, ჩვენს ინტერნეტის პროვაიდერს, შეუძლია გარკვეული CDN-ები შეგვინელოს და ამით რიგი ვებ-საიტების გამოყენება შეგვიზღუდოს. ამას გარდა, CDN-ების გამოყენებით, ჩვენი ინტერნეტ კონფიდენციალურობა ირღვევა, რადგან CDN პროვაიდერები, მაგალითდ Google, იღებს ინფორმაციას იმის შესახებ თუ რა საიტებს ვსტუმრობთ და რა სიხშირით, ხდება ჩვენზე თვალყურის დევნება (tracking) მომვალში რეკლამების ჩვენების მიზნით (Targeted Ads).

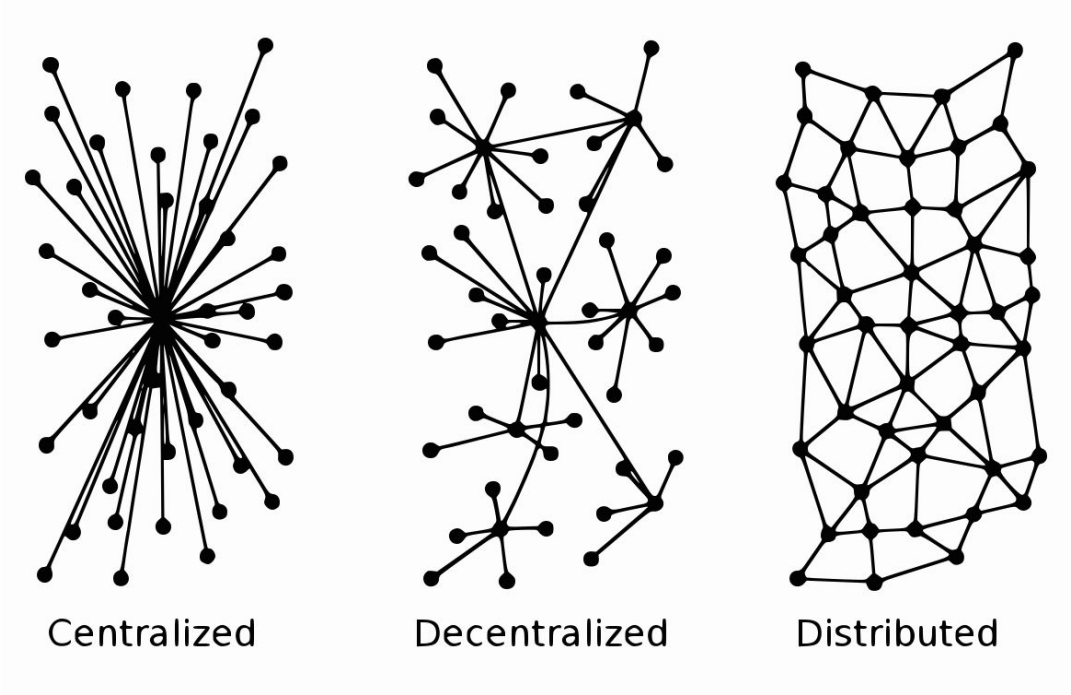
გადაჭრის გზები: ზემოთხსენებული პრობლემის გადაჭრის მრავალი გზა არსებობს, ზოგი მეცნიერი, მთლიანი ინტერნეტის ახალ მოდელს გვთავაზობს, რომელიც უფრო განვითარებულ პროტოკოლებს გამოიყენებს და მეტად მორგებული იქნება ინფორმაციის გაცვლაზე იმ გზით როგორითაც დღეს ჩვენ ამას ვაკეთებთ (ამ მეცნიერთა აზრით, დღევანდელი ინტერნეტი და HTTP პროოკოლი ფაილების, დოკუმენტების გაცვლაზეა ორიენტირებული და არა ინფორმაციის და რესურსების გაცვლაზე, ანუ როგორც ის რეალურად გამოიყენება). თეორიულად იდეა შეიძლება მოსაწონია, თუმცა მთლიანი ინტერნეტის პროტოკოლებისა და ზოგად მოდელის შეცვლა კოლოსალურ ხარჯებთან იქნება კავშირში და მრავალი დეკადა დასჭირდება, ამიტომ ეს თორია ჯერჯერობით თეორიად რჩება.

ასევე არსებობს განსხვავებული აზრი, რომლის მიხედვითაც ინტერნეტი, ბლოქჩეინის მაგვარ მოდელს უნდა იყენებდეს, ამ შემთხვევაში ყველა რესურსი დეცენტრალიზებულ იქნება და თავისუფალი ინტერნეტის პირობები არა ადამიანებისა და კომპანიების სინდისზე, არამედ იმპლემენტაციაზე, პროგრამულ რეალიზაციაზე იქნება დამოკიდებული. თუმცა ამ მოდელსაც, მსგავსი პრობლემები აქვს.

არსებობს მოსაზრება გადანაწილებული (distributed) ქსელების თაობაზე, ფსევდო-გადანაწილებულ ქსელებს დიდი კომპანიები დღესდღეობითაც იყენებენ თავიანთი

სერვერების მსოფლიოს გარშემო გასანაწილებლად, ეს მოდელი, ცენტრალიზებულსა და დეცენტრალიზებულს შორის დგას.

ზოგადად, ინტერნეტის ფუნდამენტალური ცვლილებები დღესდღეობით შეუძლებელია. აქედან გამომდინარე, ყველაზე ოპტიმალური გადაწყვეტა, რაც ამ პრობლემას შეიძლება ვუპოვოთ, იმ რესურსების, რომლებსაც ამ დიდი კომპანიებისაგან ვიღებთ, ლოკალური ასლებით ჩანაცვლებაა, რაც ჩვენს ტრაფიკს მათი სერვერებისაკენ აღარ მიმართავს და ამ სერვისების შენელება, ჩვენთვის პრობლემა აღარ იქნება.



(ცენტრალიზებული, დეცენტრალიზებული და განაწილებული ქსელები)

გადაწყვეტა: გემოთ ჩამოთვლილი იყო სამი ძირითადი ელემენტი ამ პრობლემისა:

- რეზერვირებული (Stock) ფოტოები
- ვიდეო ჰოსთინგები
- სკრიპტები CDN-ებიდან.

ფოტოებისა და ვიდეოების დეცენტრალიზება, მცირე რესურსებით შეუძლებელია, რადგან მათი უსასრულო რაოდენობა არსებობს და თითქმის ყოველი ვებ-საიტი განსხვავებულ ფოტოებსა და ვიდეოებს იყენებს. თუმცა CDN სკრიპტების რაოდენობა სასრულია, რადგან სასრულია იმ ბიბლიოთეკების რაოდენობა, რომლებსაც ვებ-გვერდების დასამზადებლად იყენებენ. ამიტომ ჩვენ შეგვიძლია აღმოვაჩინოთ ყველა მიმართვა (HTTP Request) ბროუზერიდან, დიდ CDN-ებზე, და ჩავანაცვლოთ მოთხოვნები რესურსებზე, ლოკალურად, წინასწარ ჩატვირთული რესურსებით. დავყოთ დავალება ქვე-დავალებებად:

1. ის ბიბლიოთეკები/სკრიპტები, რომლებსაც ვებ-გვერდები ხშირად ითხოვენ CDN-ებისაგან (რათა ისინი ლოკალურად შევინახოთ)
    - ამ ბიბლიოთეკების/სკრიპტების მოძიება.
    - ბიბლიოთკების/სკრიპტების შენახვა და პროგრამული უზრუნველყოფის დაყენებისას, მომხმარებლის მოწყობილობაზე შენახვა.
    - ბიბლიოთეკების/სკრიპტების მენეჯმენტი, ახალი ვერსიების დამატება, კომპილაცია, მოძველებული ვერსიების წაშლა და ა.შ.
2. წვდომა ბროუზერის გაკეთებულ მოთხოვნებზე (HTTP Request-ებზე)
    - ბროუზერისაგან საჭირო უფლებების მიღბა.
    - Request სტრუქტურის განსაზღვრა (სხვადასხვა ბროუზერს, სხვადასხვა სტრუქტურა შეიძლება ჰქონდეს)
3. ალგორითმი რომელიც დაადგენს გარკვეული მოთხოვნა უნდა დაიბლოკოს და ჩანაცვლდეს ლოკალური რესურსით, თუ არა.
    - ოპტიმიზაცია - არაეფექტური ალგორითმი გამოიწვევს მოთხოვნების შენელებას.
4. სტატისტიკის მწარმოებელი ალგორითმი, რომელიც დაითვლის თუ მოთხოვნების რა ნაწილი მოდის დიდ CDN-ებზე (ანუ რამდენად ცენტარლიზებულია ინტერნეტი)

განვიხილოთ თუ როგორ შევასრულებთ ქვე-დავალებებს:

(1) მართალია ბიბლიოთეკები და სკრიპტები ვებ-გვერდებზე ხშირად ცენტრალიზებული CDN-ებით იტვირთება, თუმცა მათი პროგრამული კოდი ხელმისაწვდომია, რადგან ასეთი ბიბლიოთეკების უდიდესი ნაწილი ღია კოდით არის მოცემული, ისეთი პოპულარული ბიბლიოთეკები როგორებიცაა: JQuery, Angular, React და ა.შ. სხვადასხვა დეველოპერების კონტრიბუციებზეა დაფუძნებული (მაგალითდ React ბიბლიოთეკა, რომელიც მართალია Facebook-ის პროდუქტია, თუმცა 1148 კონტრიბუტორი ჰყავს! იხილეთ Github ბმული: https://github.com/facebook/react). აქედან გამომდინარე მათი მოძიება არ არის რთული. ეს ბიბლიოთეკები პროგრამულ უზრუნველყოფაში იქნება ჩაშენებული და ლოკალურად ხელმისაწვდომი ნებისმიერ დროს.

რაც შეეხება განახლებებს და ვერსიების კონტროლს, ეს შედარებით რთული ამოცანაა, პროგრამის პირველ ვერსიაში ამ ყველაფრის გაკეთება, დეველოპერებს ხელით მოგვიწევს, თუმცა მომავალში შესაძლებელია Github-ის მეშვეობით განახლებების გამოწერა და ამ პროცესის ავტომატიზაცია.

(2) ბროუზერის მიერ გაკეთებულ მოთხოვნებზე წვდომის მიღება შეგვიძლია იმ შემთხვევაში თუ ჩვენს პროგრამულ უზრუნველყოფას როგორც ბროუზერის გაფართოებას ისე წარმოვადგენთ, ამ შემთხვევაში ის ხელმისაწვდომი იქნება ბროუზერების

გაფართოებების კატალოგშიც. (მაგ. ქრომისათვის: https://chrome.google.com/webstore/category/extensions)

მოთხოვნის (Request) სტრუქტურა შეგვიძლია მოვიძიოთ ვებ სტანდარტების სპეციფიკაციებში (იხ. https://www.w3.org/Protocols/rfc2616/rfc2616-sec5.html), რადგან მთლიანი სტრუქტურა ჩვენთვის მნიშვნელოვანი არ არის, ის შეგვიძლია მარტივი სახით ასე წარმოვადგინოთ:

```
abstract class Request {
  string method;
  string URI;
  Map<string, string> headers;
  Map<string, string> body;
  /* და ა.შ. სხვა ველები */
}
```

ეს არის აბსტრაქტული კლასი რომლის იმპლემენტაცია სხვადასხვა ბროუზერისთვის სხვადასხვა კონკრეტული კლასი შეიძლება იყოს.

აქ მთავარი ინფორმაცია, რომელიც გვაინტერესებს არის URI ველი (ანუ რესურსის მისამართი), მასში წერია თუ რა რესურზე ვაკეთებთ მოთხოვნას და რომელ CDN-ს მივმართავთ.

(3) იმისათვის რათა დავადგინოთ, ესა თუ ის მოთხოვნა, უნდა ჩანაცვლდეს ლოკალური რესურსით თუ არა, საჭიროა:

- მოთხოვნის ობიექტი (იხ. წინა პუნქტი) - აღვნიშნოთ *req* ცვლადით
- სია იმ CDN-ებისა რომელი გვინდა რომ ჩავანაცვლოთ - საწყისი სია შედგება: Google, Facebook, Microsoft, Amazon, Yandex, Baidu კომპანიების CDN-ებისაგან, თუმცა მომავალში ის უფრო და უფრო გაიზრდება Open source კონტრიბუციების მეშვეობით - აღვნიშნოთ List<string> cndList ცვლადით.
- სია იმ ბიბლიოთეკებისა რომლებიც ლოკალურად არის ხელმისაწვდომი - აღვნიშნოთ Map<Pair<string, string>, string> libraryList ცვლადით, შეიცავს ბიბლიოთეკის დასახელებას და ვერსიას (key) და ბიბლიოთეკის ადგილმდებარეობას დისკზე (value).
- ნიმუშების (Pattern) სია, რაგან ყველა CDN-ს განსხვავებული მიმართვის URI აქვს, საჭიროა წინასწარ ვიცოდეთ ყველა შესაძლო ნიმუში. ნიმუშის მაგალითია:
  https://cdn.com/libs/${name}/${version}/umd/${name}.js

როგორც ხედავთ ეს ნიმუში ორ ცვლადს: name - ბიბლიოთეკის სახელს და version - ბიბლიოთეკის ვერსიას შეიცავს, ამ ცვლადების აღმოჩენა რეგულარული გამოსახულებების (Regular expressions) მეშვეობით შეგვიძლია. შევინახოთ ნიმუშების სია ცვლადში List<Pattern> patternList, სადაც Pattern კლასი არი შემდეგნაირი:

```
class Pattern {
  class ParseResult {
    string libraryName;
    string libraryVersion;
  }

  // თუ URI-ში არმოჩნდა ბიბლიოთეკა, აბრუნებს
  // ParseResult ობიექტს რომელიც შეიცავს
  // ბიბლიოთეკის სახელსა და ვერსიას,
  // სხვა შემთხვევაში null
  ParseResult check(string uri);
}
```

სადაც, check მეთოდი ამოწმებს, მოცემული მისამართი (uri) არის თუ არა CDN-ის და შეიცავს თუ არა ბიბლიოთეკის/სკრიპტის სახელსა და ვერსიას.

მას შემდეგ რაც ყველა ზემოთ ჩამოთვლილი ცვლადი და სტრუქტურა გვაქვს, Interceptor (მოთხოვნების დამამუშავებელი) მოდულის ალგორითმი (ფსევდო კოდი) მსგავსად გამოიყურება:

ყოველი მოთხოვნისათვის req, დააპაუზე მოთხოვნა და გააკეთე შემდეგი:

a. ყოველი ნიმუშისათვის pattern სიიდან patternList გამოიძახე pattern.check(req.uri), თუკი აღმოჩენილია ბიბლიოთეკა და cdn, გადადი b პუნქტზე, თუ არ არის აღმოჩენილი გადადი c პუნტქზე.
b. მოძებნე ნაპოვნი ბიბლიოთეკა და ვერსია libraryList ცვლადში,
    i. თუ ნაპოვნია გადადი d პუნქტზე,
    ii. თუ არ არის ნაპოვნი, გადადი e პუნქტზე

c. რადგან ბიბლიოთეკა არ არის აღმოჩენილი,
    i. განაგრძე მოთხოვნა,
    ii. გამოიძახე სტატისტიკის მოდული (აღწერილია შემდგომში)
    iii. გადადი შემდეგი მოთხოვნის განხილვაზე.
d. რადგან ბიბლიოთკა და ვერსია აღმოჩენილია ლოკალურად
    i. შეწყვიტე მოთხოვნა და ჩაანაცვლე ის ლოკალური რესურსით.
    ii. გამოიძახე სტატისტიკის მოდული.
    iii. გადადი შემდეგ მოთხოვნაზე.
e. რადგან ბიბლიოთეკა ნაპოვნია მოთხოვნის მისამართში, თუმცა ლოკალურად ეს ბიბლიოთეკა ან მისი ეს ვერსი არ გვაქვს
    i. განაგრძე მოთხოვნა და ჩაიწერე ის CDN-იდან.
    ii. დაამატე CDN-იდან გადმოწერილი რესურსი, რესურსების სიაში.
    iii. ჩაინიშნე ლოგი იმის შესახებ რომ ეს ბიბლიოთეკა/ვერსია დასამატებელია
    iv. გამოიძახე სტატისტიკის მოდული.

შემდეგნაირად გამოიყურება სტატისტიკის მოდული:

```
enum Status {
  NOT_CDN,
  NOT_FOUND_LOCALY,
  REPLACED,
}

interface Statistics {
  void report(Status x, string libraryName, string libraryVersion);
}
```

ის ინახავს ინფორმაციას იმის შესახებ თუ როგორ ნაწილდება ჩვენი ტრაფიკი. სხვა მოდულები მასთან ურთიერთქმედებენ report მეთოდის მეშვეობით.

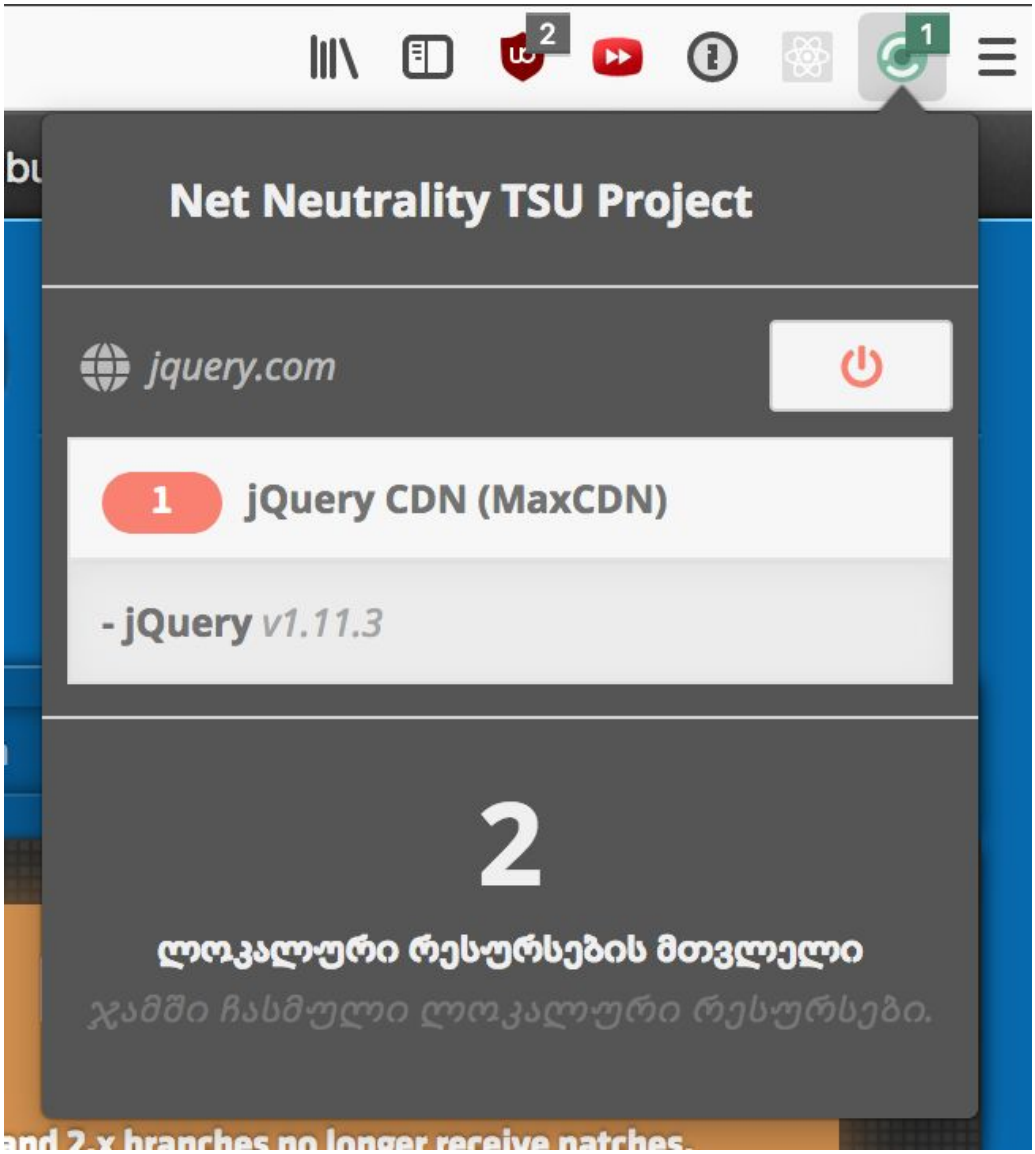
ანალიზი: მართალია ჩვენი პროგრამული უზრუნველყოფა ზემოთ მოცემული პრობლემის მხოლოდ ნაწილს წყვეტს, თუმცა CDN-ები და მათ მიერ მოწოდებული ბიბლიოთეკები/სკრიპტები, ამ პრობლემის უდიდესი ნაწილია, რადგან ისინი შეიცავენ ზუსტად იმ კოდს რომელიც ჩვენს ბროუზერში ეშვება და ბევრად მეტ კონტროლს აძლევენ CDN-ებს ჩვენს მოწყობილობებზე, ვიდრე მაგალითად ვიდეო და ფოტო ჰოსთინგები.

თუმცა ეს გადაწყვეტაც, რა თქმა უნდა, გარკვეულ საფასურს მოითხოვს:

- იმისათვის რომ პროგრამული უზრუნველყოფა გამოვიყენოთ, საჭიროა ის ჩვენს ბროუზერში დავაყენოთ, რაც გარკვეულ მეხსიერებას მოითხოვს ჩვენს მოწყობილობაზე.
- ასევე რადგან გვიწევს ზოგი მოთხოვნის დაყოვნება, შესაძლოა ზოგი მოთხოვნა (უმნიშვნელოდ თუმცა მაინც) შენელდეს.
- ასევე საჭიროა ხშირი განახლებები, რათა ჩვენი ლოკალური ბიბლიოთეკები არ ჩამორჩეს CDN-ებზე არსებულ ვერსიებს

მეორე მხრივ, უპირატესობები არის:

- დეცენტრალიზებული ბიბლიოთეკები/სკრიპტები
- ინტერნეტ სერვისის პროვაიდერი ვერ შეძლებს ჩვენს შენელებას
- CDN პროვაიდერები ვერ შეძლებენ ჩვენზე თვალყურის დევნებას (Tracking)
- დავზოგავთ ინტერნეტ ტრაფიკს, რადგან არ მოგვიწევს CDN-ებიდან რესურსების გადმოწერა, ისინი წინასწარ ერთჯერადად გვექნება ლოკალურად.



(სქრინშოტი პროგრამული უზრუნველყოფიდან)

გამოყენებული ლიტერატურა

- Paid Prioritization and Its Impact on Net Neutrality, IEEE Journal on Selected Areas in Communications
- Crowcroft, Jon (2007). Net Neutrality: The Technical Side of the Debate: A White Paper (PDF). University of Cambridge.
- Meza, Philip E. (20 March 2007). Coming Attractions?. Stanford University Press. p. 158
- "Open Internet". Federal Communications Commission. Archived from the original on 26 May 2017. Retrieved 26 May 2017.
- https://blog.parse.ly/post/2855/facebook-continues-to-beat-google-in-sending-traffic-to-top-publishers/
- https://www.statista.com/statistics/346167/facebook-global-dau/
- https://www.statista.com/chart/4298/amazons-long-term-growth/