

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Tanel Tomson

Töö pealkiri täpsustada

Visualizing network switches

Bakalaureusetöö (9 EAP)

Juhendaja: Meelis Roos, MSc

Tartu 2018

Võrgukommutaatorite visualiseerimine

Lühikokkuvõte:

One or two sentences providing a basic introduction to the field, comprehensible to a scientist in any discipline.

Two to three sentences of more detailed background, comprehensible to scientists in related disciplines.

One sentence clearly stating the general problem being addressed by this particular study.

One sentence summarising the main result (with the words “here we show” or their equivalent).

Two or three sentences explaining what the main result reveals in direct comparison to what was thought to be the case previously, or how the main result adds to previous knowledge.

One or two sentences to put the results into a more general context.

Two or three sentences to provide a broader perspective, readily comprehensible to a scientist in any discipline, may be included in the first paragraph if the editor considers that the accessibility of the paper is significantly enhanced by their inclusion.

Võtmesõnad:

List of keywords

CERCS:

CERCS kood ja nimetus: <https://www.etis.ee/Portal/Classifiers/Details/d3717f7b-bec8-4cd9-8ea4-c89cd56ca46e>

Visualizing network switches

Abstract:

Abstract

Keywords:

List of keywords

CERCS:

CERCS code and name: <https://www.etis.ee/Portal/Classifiers/Details/d3717f7b-bec8-4cd9-8ea4-c89cd56ca46e>

Sisukord

1	Sissejuhatus	7
2	Sarnased lahendused	8
3	Rakenduse kirjeldus	9
3.1	Kasutamise eeldused	9
3.2	Rakenduse nõuded	9
3.3	Nõuete täitmine	10
4	Kasutatud tehnoloogiad	11
4.1	CDP ja LLDP	11
4.2	SNMP	11
4.3	easySNMP	12
4.4	Cytoscape.js	12
5	Valminud rakenduse kirjeldus	13
5.1	Tagasüsteem	13
5.1.1	Rakenduse seadistamine	13
5.1.2	Rakenduse põhivoog	14
5.1.3	Päritavad andmed	14
5.1.4	Andmete faili salvestamine	15
5.1.5	Veahaldus	15
5.1.6	Logimine	15
5.2	Kasutajaliides	15
5.2.1	Välised teegid	15
5.2.2	Lokaalse faili laadimine brauseris	16
5.2.3	Tagasüsteemi loodud sisendi lugemine	16
5.2.4	Graafi kuvamine	17
5.2.5	Kasutaja muudatuste salvestamine	17
6	Edasised tegevused	18
7	Conclusion	19
	Viidatud kirjandus	20
	Lisad	21
	I. Glossary	21
	II. Litsents	22

Unsolved issues

Töö pealkiri täpsustada	1
One or two sentences providing a basic introduction to the field, comprehensible to a scientist in any discipline.	2
Two to three sentences of more detailed background, comprehensible to scientists in related disciplines.	2
One sentence clearly stating the general problem being addressed by this particular study.	2
One sentence summarising the main result (with the words “here we show” or their equivalent).	2
Two or three sentences explaining what the main result reveals in direct comparison to what was thought to be the case previously, or how the main result adds to previous knowledge.	2
One or two sentences to put the results into a more general context.	2
Two or three sentences to provide a broader perspective, readily comprehensible to a scientist in any discipline, may be included in the first paragraph if the editor considers that the accessibility of the paper is significantly enhanced by their inclusion.	2
List of keywords	2
CERCS kood ja nimetus: https://www.etis.ee/Portal/Classifiers/Details/d3717f7b-bec8-4cd9-8ea4-c89cd56ca46e	2
Abstract	2
List of keywords	2
CERCS code and name: https://www.etis.ee/Portal/Classifiers/Details/d3717f7b-bec8-4cd9-8ea4-c89cd56ca46e	2
Eemalda lõplikust versioonist	6
What is it in simple terms (title)?	7
Why should anyone care?	7
What was my contribution?	7
What you are doing in each section (a sentence or two per section)	7
Mingi sissejuhatus võrgutehnoloogiasse; võrgukommutaatorite selgitus	7
Sissejuhatav tekst	8
Loetelu neist rakendustest	8
Netdisco	8
netcrawl	8
Riistvara ja tarkvara jm nõuded	9
Täpsustada, mis andmeid SNMP peab jagama	9
CDP kas mõistetes või lehekülje alla	9
Eraldada funktsionaalsed ja mittefunktsionaalsed?	9
Nimekiri üle käia	9

Kirjeldada rakendust ning näidata nõuete täitmist	10
Kas siin peaks git ka olema?	11
Kas siin peaks python ka olema?	11
See siin lõhnab kantseliidi järele - oluline enne, laialdaselt kasutatav; lihtsasti toimiv on mõttetus	11
faktikontroll ja viited	11
halb sõnastus - samade id-de	11
Community string	11
faktikontroll ja viited	11
täpsustada 1 vs 2	11
täpsustada, kas 2 või 2c vm	11
põhjendada, miks 2	11
faktikontroll ja viited	11
veenduda, et kirjeldab ka	12
Kas rääkida teegi kaladest? https://github.com/kamakazikamikaze/easysnmp/issues/69	12
Siin veel vaja mõelda, kuidas seda kirja panna	14
Kas teha lihtne pseudokood?	14
LLDP ja CDP mõlemat kasutatakse, kumba eelistatakse mõlema töötamise korral?	14
Mis andmeid küsitakse	14
Mil viisil easySNMP-ga andmeid küsitakse	14
sisemiselt kasutatakse unikaalsuse identifikaatoriks sysName	14
Tabel OID-dest ja kus neid andmeid kasutatakse?	14
Iga OID kohta: oid, kas seadme kohta või naabri?, kus kasutajaliideses kuvatakse?	14
Itereeritakse andmeid ja kirjutatakse faili, mis on ühtlasi .js fail	15
Sissejuhatus kasutajaliidese kohta	15
kasutatakse cytoscape'i, mis failid includedud	17
cytoscape'i implementatsioon ja konf	17
puudus graafi servade labe'itega - või see äkki puuduste alla?	17
nõuded ja nende implementeerimine	17
räsi arvutamine ja teate kuvamine, kui andmefail on taustal muutunud, kui see implementeeritud saab	17
Sissejuhatav tekst	18
what did you do?	19
What are the results?	19
future work?	19
kuupäev	22

Eemalda lõplikust versioonist

1 Sissejuhatus

What is it in simple terms (title)?

Why should anyone care?

What was my contribution?

What you are doing in each section (a sentence or two per section)

Mingi sissejuhatus võrgutehnoloogiasse; võrgukommutaatorite selgitus

Tip: if it's hard for you to start writing, then try to split it to smaller parts, e.g. if the title is "Type Inference for a Cryptographic Protocol Prover Tool" then the "What is it" can be divided into "what is type inference", "what is cryptographic protocol" and "what is the prover tool". These three can also be split to smaller parts etc.

2 Sarnased lahendused

Sissejuhatav tekst

Käesolevas peatüki eesmärk on tutvustada ja anda ülevaade sarnastest lahendustest, tuues välja nende tugevused ja puudused.

Loetelu neist rakendustest

Netdisco

netcrawl

<https://github.com/ytti/netcrawl>

3 Rakenduse kirjeldus

Enne rakenduse programmeerimist fikseeriti rakenduse nõuded. Järgnev peatükk kirjeldab eeldused rakenduse käivitamiseks ning selle nõuded.

Riistvara ja tarkvara jm nõuded

3.1 Kasutamise eeldused

Täpsustada, mis andmeid SNMP peab jagama

CDP kas mõistetes või lehekülje alla

Rakenduse kasutamiseks on tehtud eeldus, et kasutajal on kontroll kaardistatavate võrgukommutaatorite üle. Võrgukommutaatorid peavad olema programmi jaoks kättesaadavad ja vajalikku informatsiooni edastama. See tähendab, et võrgukommutaatoritel peab töötama SNMP teenus ja see peab olema seadistatud jagama CDP andmeid.

3.2 Rakenduse nõuded

Eraldada funktsionaalsed ja mittefunktsionaalsed?

Nimekiri üle käia

1. Rakenduse käivitamine toimub käsurealt, vajalik python (koos vähemalt ühe SNMP teegiga) ja SNMP
2. Rakenduse sisendiks on ühe või mitme võrgukommutaatori aadress ja seadmetes seadistatud SNMP kommuuni nimi
3. ? Autodiscovery koos piiridega - võimalus anda ette võrgu piirid, kust otsida
4. Kui antakse ette mitu seadet, on võimalus määrata erinevatele seadmetele erinevad SNMP kommuunide nimed
5. Rakendus küsib igalt sisendiks saadud seadmelt LLDP/CDP andmed (nimi, ip, mac; ühenduste füüsiline port, nimi, ip, mac)
6. Kui seadmega on ühendatud veel kommutaatoreid, küsitakse ka neilt andmed
7. Rakendus töötleb ja vormindab saadud andmed
8. Tekkinud vead kuvatakse kasutajale mõistlikult (mis juhtus, mida teha)
9. Rakendus loob antud andmete põhjal staatilise veebilehe (.html), kus kuvatakse graaf võrgukommutaatorite kohta

10. Staatilise veebilehe vaatamiseks pole vaja internetiühendust (CSS ja JS lokaalsed)
11. Graafi tipud on võrgukommutaatorid
12. ? Graafi väiksemad (vähem silmapaistvavad) tipud on võrgukommutaatoritesse ühendatud seadmed (mis pole kommutaatorid)
13. ? Või siis vaikimisi peidus ja peab lahti klikkima
14. Graafi servad tippude vahel on võrgukommutaatorite ühendused, erinevad ühenduste kiirused on tähistatud erinevalt
15. Graafis on vaikimisi paigutus mõistlik, see tähendab, et tipud ei kattu ja kui tekib mitu graafi, kuvatakse need üksteisest pisut eemal
16. Graafis on võimalik seadmeid enda suva järgi lohistada
17. Graafis kuvatakse tippudes seadme nimi (aadress), aga peale vajutades on võimalik näha ka ülejäänud infot (ip, mac, port)
18. Kasutaja graafile tehtud muudatused salvestatakse kasutaja brauseriga lokaalselt (küpsised või localStorage) ja taastatakse järgmisel avamisel

3.3 Nõuete täitmine

Kirjeldada rakendust ning näidata nõuete täitmist

4 Kasutatud tehnoloogiad

Antud peatükis tutvustatakse tehniliselt võrguprotokolle, teeke ning muid vahendeid, mida valminud rakendus kasutab.

Kas siin peaks git ka olema?

Kas siin peaks python ka olema?

HTML, JS, CSS

4.1 CDP ja LLDP

CDP (*Cisco Discovery Protocol*) ja LLDP (*Link Layer Discovery Protocol*) on mõlemad võrguprotokollid ühendatud võrguseadmetele haldusinformatsiooni jagamiseks. LLDP ja CDP on funktsionaalsuse poolest küllaltki sarnased. Peamine erinevus kahe protokollide vahel on, et CDP on Cisco (võrguseadmete tootja) loodud ja hallatav omand-protokoll. [Bha15a] LLDP seevastu on avatud ja seadmetootjate ülene protokoll, LLDP standardit haldab IEEE. [Bha15b]

4.2 SNMP

See siin lõhnab kantseliidi järele - oluline enne, laialdaselt kasutatav; lihtsasti toimiv on mõttetus

SNMP (*Simple network management protocol*) on laialdaselt kasutatav ja lihtsasti toimiv võrguhaldusprotokoll, mis kuulub OSI mudelis rakenduskihti. Protokoll lihtsustab ja ühtlustab võrguseadmetelt haldusinfo küsimist. [Laa08, 151]

faktikontroll ja viited

Hallatavad seadmed hoiavad haldusinfot MIB-i struktuuris, mis on hierarhiline andmebaas. Igal objektil on oma id (OID). Samade id-de abil käib ka seadmetelt andmete küsimine.

halb sõnastus - samade id-de

Community string

faktikontroll ja viited

SNMP-st on 3 põhilist versiooni: 1, 2 ja 3.

täpsustada 1 vs 2

Valminud rakendus kasutab versiooni 2.

täpsustada, kas 2 või 2c vm

põhjendada, miks 2

faktikontroll ja viited

Valminud rakendus kasutab SNMP protokoll, et sisendiks saadud võrguseadmetelt CDP ja LLDP andmeid küsida. Täpsemalt kirjeldatakse seda peatükis 4.3.

veenduda, et kirjeldab ka

4.3 easySNMP

Kommutaatoritelt üle SNMP protokoll andmete küsimiseks kasutab rakendus Pythoni teeki easySNMP. Töö algfaasis katsetati alternatiivina ka teeki PySNMP¹. Valik tehti aga easySNMP kasuks, kuna PySNMP on kommutaatoritega suhtlemisel tuntavalt aeglasem. [eas]

Kas rääkida teegi kaladest? <https://github.com/kamakazikamikaze/easysnmp/issues/69>

4.4 Cytoscape.js

Järgmiseks oli vaja Javascripti teeki graafide brauseris esitamiseks. Andmeid visualiseerivaid (ja muuhulgas graafe toetavaid) teeki leidub arvukalt, ent autor kitsendas otsingu lihtsuse huvides vaid graafidele spetsialiseerunud teekidele. Samuti pidi teek olema avatud lähtekoodiga ning toetama graafi tippudele hüpikvihjete lisamist.

Kasutusele võeti Cytoscape.js², mis vastas kirjeldatud nõuetele. [cyt] Pärast esmaseid katsetusi demodega tõusis esile hea dokumentatsioon, mis tegi teegi kasutamise lihtsaks, lai valik erinevaid kujundusi ning graafide kuvamisel mõistlik ekraanipinna kasutamine.

Kuigi teek vaikimisi hüpikvihjeid kuvada ei oska, on selleks loodud laiendeid. Hüpikvihjete kuvamiseks kasutusele teegid on kirjeldatud peatükis 5.2.1.

¹<http://snmplabs.com/pysnmp>

²<http://js.cytoscape.org/>

5 Valminud rakenduse kirjeldus

Rakenduse võib funktsionaalsuse järgi jagada kaheks: tagasüsteem ja kasutajaliides. Tagasüsteemi ülesanne on küsida ja töödelda sisendiks saadud võrguseadmetelt info ühendatud seadmete kohta. Kasutajaliides koostab tagasüsteemi poolt saadud andmetest veebilehe kommutaatorite graafiga. Peatükis kirjeldatakse detailselt mõlema osa arhitektuuri.

5.1 Tagasüsteem

Tagasüsteem on kirjutatud programmeerimiskeeles Python. Välisest teekidest on kasutusel vaid easySNMP (4.3). Tagasüsteemi lähtekood asub `src/` kaustas (va `src/web/`, mis on kasutajaliides).

5.1.1 Rakenduse seadistamine

Rakenduse sätted loeb programm failist `config.ini`. Rakenduse lähtekoodis on näidisfail `config.ini.sample`, mille rakenduse käivitaja saab kopeerida ja vastavalt soovidele muuta.

```
[Application Config]

# Default community string is used when switch has no specific
  community string set
defaultCommunityString = public

# Switches to be looked up.
switches = with-specific-community.example.com specificcommunity
           with-default-community.example.com
           another-with-default-community.example.com

# Enable or disable application logging (true or false)
debug = false
```

Joonis 1. Näidis sättefail `config.ini.sample`

Kasutaja saab määrata kommutaatorid, millelt naabrite info küsitakse. Seejuures saab igale kommutaatorile vajadusel määrata eraldi kogukonnasõne (vt 4.2 ja samuti nõue 4). Lisaks on võimalik seadistada rakenduse logimist silumise tasemele (täpsemalt vt 5.1.6).

Rakendus kasutab sätetefaili parsimiseks Pythoni moodulit `configparser`³ ja sellest lähtuvalt on sätetefail INI-struktuuriga. Sätetefaili puudumisel või kui sätete lugemine

³<https://docs.python.org/3.6/library/configparser.html>

ebaõnnestub, kuvatakse kasutajale vastav veateade ja programm lõpetab töö. Sätete lugemise implementatsioon on failis `src/config_helper.py`.

5.1.2 Rakenduse põhivoog

Kui sätetefailist on seadistatud kommutaatorid loetud, asutakse neilt andmeid küsima. Seda tehakse ükshaaval, üle kommutaatorite itereerides. Vastuseks saadud andmed hoiustatakse sisemiselt sõnastikus, seejuures jagatakse iga kommutaatori käest saadud andmed kolmeks:

1. kommutaatori enda andmed (graafis tipud)
2. kommutaatoriga ühendatud teiste kommutaatorite andmed (graafis servad)
3. ülejäänud kommutaatoriga ühendatud seadmete info.

Siin veel vaja mõelda, kuidas seda kirja panna

Iga saadud naabri korral kontrollitakse, kas see on seadistatud kommutaator (ehk graafis teine tipp). Kui jah, siis talletatakse andmed, mis on vajalikud graafi serva kuvamiseks (nt pesade nimed, ühenduse kiirus). Samas sel juhul ei kasutata naabri kohta saadud andmeid. Teisisõnu - eelistatakse kommutaatori andmeid, mida kommutaator ise enda kohta andis, mitte andmeid, mida temaga ühenduses olev kommutaator tema kohta andis. Kui naaber aga ei ole teine sisendiks saadud kommutaator (vaid kommutaatoriga ühendatud muu seade), siis on tema kohta saadud info meile oluline ja naabri info salvestatakse.

Kas teha lihtne pseudokood?

LLDP ja CDP mõlemat kasutatakse, kumba eelistatakse mõlema töötamise korral?

Mis andmeid küsitakse

Mil viisil easySNMP-ga andmeid küsitakse

5.1.3 Päritavad andmed

Kommutaatorite käest küsitavad andmed võib jagada kaheks: info seadme enese kohta ja info kommutaatoriga ühenduses olevate seadmete (naabrite) kohta. Järgnevalt toome välja kõik andmed, mida kommutaatoritelt päritakse, sealhulgas ka SNMP identifikaatorid (OID-d, vt 4.2), mida päringutes kasutati.

sisemiselt kasutatakse unikaalsuse identifikaatoriks `sysName`

Tabel OID-dest ja kus neid andmeid kasutatakse?

Iga OID kohta: oid, kas seadme kohta või naabri?, kus kasutajaliideses kuvatakse?

5.1.4 Andmete faili salvestamine

Itereeritakse andmeid ja kirjutatakse faili, mis on ühtlasi .js fail

Põhjus, miks kasutatakse Javascripti faili (laiendiga .js) ja mitte näiteks JSON-formaati, on kirjeldatud peatükis 5.2.3.

Andmete faili kirjutamise implementatsioon on failis `src/output_helper.py`.

5.1.5 Veahaldus

Vigadele kõige altim on kommutaatoritelt andmete küsimine - juhul kui kommutaatoriga ei saada ühendust (vale seadistus, võrguprobleem vms). Veahalduse roll on sel juhul tagada, et programmi töö jätkuks teistelt kommutaatoritelt andmete küsimisega. Sel juhul logitakse kasutajale olemasolevate andmetega võimalikult informatiivne viga ja minnakse programmi vooga edasi.

5.1.6 Logimine

Logimiseks kasutatakse Pythoni moodulit `logging`⁴. Rakenduse sättefailis on võimalik määrata logimise taset (`debug`, kui `debug = true` või `info`, kui `debug = false`). Debug tasemel logikirjed on mõeldud vaid arendamiseks ja vigade silumiseks.

5.2 Kasutajaliides

Kasutajaliides on staatiline veebileht.

Sissejuhatus kasutajaliidese kohta

Kasutajaliidese lähtekood asub `src/web/` kataloogis.

5.2.1 Välised teegid

Graafide kuvamiseks kasutatakse Javascripti teeki `cytoscape.js` (4.4). Lisaks kasutatakse laiendit `cytoscape-popper`⁵, mis lisab teegile `Popper.js`⁶ (teek muuhulgas hüpikvihjete kuvamiseks) toe. Lisaks võeti kasutusele `Tippy.js`⁷, mis teeb hüpikvihjete loomise ja kohandamise `Popper.js`-iga lihtsamaks.

Välised teegid asuvad kataloogis `src/web/lib/`. Kõik kasutatud välised failid on lähtekoodiga kaasas (ei laeta internetist). Põhjuseks turvalisus ja võimalus rakendust ilma võrguühendusega kasutada.

⁴<https://docs.python.org/3.6/library/logging.html>

⁵<https://github.com/cytoscape/cytoscape.js-popper>

⁶<https://popper.js.org>

⁷<https://atomiks.github.io/tippyjs/>

5.2.2 Lokaalse faili laadimine brauseris

Lokaalsel veebilehel lokaalse faili Javascriptiga lugemine on keerulisem kui võiks arvata. Brauserid kaitsevad kasutajaid pahatahtliku koodi laadimise ja jooksumise eest, aga samamoodi piiratakse ka pääsu kasutaja lokaalsetele failidele.

Brauserid rakendavad reeglina doomenisisese ressursikasutuse (*same-origin policy*) printsiipi. See tähendab, et veebileht ei saa jooksumata koodi ning ei pääse ligi failidele, mis asuvad teistes domeenides. Et võimaldada ka doomenivälist ressursikaasutust on W3C loonud spetsifikatsiooni CORS (*Cross-origin resource sharing*). [w3C14]

Antud spetsifikatsiooni implementeerivad brauserid ise ning sellele on erinevad arendajad lähenenud erinevalt [Bar08]. Loodud programmi kontekstis tuli esile probleem, et kasutades `file://` URI-skeemi, on faili päritolu (*origin*) definitsioon CORS-i spetsifikatsioonis lahtine [Bar11, peatükk 4, punkt 4]. See tähendab, et brauserid käituvad antud olukorras erinevalt ja ei ole garantiid, et igas brauseris tagasüsteemi poolt loodud faili lugemine õnnestub.

5.2.3 Tagasüsteemi loodud sisendi lugemine

Töö autor lootis algselt kommutaatorite andmed tagasüsteemilt kasutajaliidesele edastada JSON-formaadis, seejuures kasutada Pythoni ja Javascripti standardteeke. Katsetuste käigus tuli eelnevalt kirjeldatud erinev brauserite käitumine aga välja. Chromium (ja Google Chrome) ei lubanud `file://` URI-skeemiga lokaalset faili üldse laadida. Firefox lubaks faili laadida, aga (alates küljendusmootori Gecko versioonist 1.9) seab ette nõuded faili asukohale [ffC].

Kasutaja brauserite kohta eeldusi teha ei tahetud - programm peaks töötama sõltumata brauserist. Üks lahendus oleks veebirakenduse kuvamiseks kasutada lokaalset veebiserverit ja küsida andmefail üle `http://` URI-skeemi^{8 9 10}. See on aga töö raames loodava rakenduse kontekstis üleliigne keerukus ja probleemi lahenduseks ei sobinud.

Selle asemel otsustati andmed tagasüsteemi poolt vormindada Javascripti koodifailiks. Fail defineerib ühe muutuja, milles on kommutaatorite andmed sõnastikus. Seejuures vormindatakse andmed sõnastikus nõnda, et selle saab otse cytoscape.js (vt 4.4) teegile ette anda. See tähendab, et sõnastikus on elemendid `nodes` ja `edges`, vastavalt graafi tipud ja servad. Antud fail laetakse kasutajaliideses Javascripti lähtekoodina ja seejärel on kommutaatorite andmed kasutajaliideses olemas ja neid saab kasutajale kuvada.

⁸<https://stackoverflow.com/questions/46258449/cors-error-requests-are-only-supported-for-protocol-schemes-http-etc>

⁹<https://stackoverflow.com/questions/20041656/xmlhttprequest-cannot-load-file-cross-origin-requests-are-only-supported-for-ht>

¹⁰<https://stackoverflow.com/questions/35335047/how-to-get-rid-of-cross-origin-request-block-in-chrome>

5.2.4 Graafi kuvamine

kasutatakse cytoscape'i, mis failid includedud

cytoscape'i implementatsioon ja konf

puudus graafi servade labe'itega - või see äkki puuduste alla?

nõuded ja nende implementeerimine

5.2.5 Kasutaja muudatuste salvestamine

Vastavalt nõudele 18 pidi kasutajaliides salvestama kasutaja muudatused graafile (tippude asukohtade muudatused). Antud funktsionaalsust teek cytoscape.js otse ei paku, küll aga defineeritakse funktsioon `cy.json()`¹¹, mis võimaldab graafi paigutust JSON-formaadis importida ja eksportida.

Implementeeriti graafi asetuse salvestamine, kui kasutaja veebilehelt lahkub või seda värskendab. JSON andmed salvestatakse sõnena brauseri lokaalsesse andmesalvestisse (`localStorage`¹²). Kui kasutaja veebilehe avab, laetakse ja taastatakse graafi paigutus andmesalvestist. Kui seda varasemalt salvestatud ei ole (esimene külastus), siis laetakse graaf andmefailist.

Samuti lisati veebilehele nupp graafi paigutuse taastamiseks - graafi andmefailist uuesti laadimiseks.

räsi arvutamine ja teate kuvamine, kui andmefail on taustal muutunud, kui see implementeeritud saab

¹¹<http://js.cytoscape.org/#cy.json>

¹²<https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>

6 Edasised tegevused

Sissejuhatav tekst

7 Conclusion

what did you do?

What are the results?

future work?

Viidatud kirjandus

- [Bar08] Adam Barth. Security in depth: Local web pages. Chromium Blog, <https://blog.chromium.org/2008/12/security-in-depth-local-web-pages.html>, 2008.
- [Bar11] Adam Barth. The web origin concept. <https://tools.ietf.org/html/rfc6454>, 2011.
- [Bha15a] Deben Bhattarai. Cisco discovery protocol (cdp). The Cisco Learning Network, 2015. <https://learningnetwork.cisco.com/docs/DOC-26872>.
- [Bha15b] Deben Bhattarai. Link layer discovery protocol (lldp). The Cisco Learning Network, 2015. <https://learningnetwork.cisco.com/docs/DOC-26851>.
- [cyt] Cytoscape.js. <http://js.cytoscape.org/#introduction>. Vaadatud: 25.03.2019.
- [eas] Easy snmp documentation. <https://easysnmp.readthedocs.io>. Vaadatud: 25.03.2019.
- [ffC] Same-origin policy for file: Uris. https://developer.mozilla.org/en-US/docs/Archive/Misc_top_level/Same-origin_policy_for_file:_URIs. Vaadatud: 26.03.2019.
- [Laa08] Erkki Laaneoks. *Sissejuhatus võrgutehnoloogiasse*. Tartu Ülikooli Kirjastus, 2008.
- [w3C14] Cross-origin resource sharing. W3C Recommendation, <https://www.w3.org/TR/cors/>, 2014.

Lisad

I. Glossary

II. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, **Tanel Tomson**,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

Võrgukommutaatorite visualiseerimine

mille juhendaja on Meelis Roos

- 1.1 reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2 üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

kuupäev

Tartus, pp.kk.aaaa