

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matej Rojko

**Rešitev za izris omrežne topologije v srednjih in velikih računalniških
omrežjih**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2011

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matej Rojko

**Rešitev za izris omrežne topologije v srednjih in velikih računalniških
omrežjih**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

Mentor: doc. dr. Zoran Bosnić

Ljubljana, 2011

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.



Št. naloge: 00127/2011

Datum: 15.06.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **MATEJ ROJKO**

Naslov: **REŠITEV ZA IZRIS OMREŽNE TOPOLOGIJE V SREDNJIH IN VELIKIH
RAČUNALNIŠKIH OMREŽJIH**
**SOLUTION FOR NETWORK TOPOLOGY VISUALIZATION IN MEDIUM-
SIZED AND LARGE COMPUTER NETWORKS**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

V računalniških omrežjih z velikim številom naprav je skrbnikom omrežja moč olajšati delo z vizualizacijo omrežne topologije. Zaradi velikega števila priključenih naprav in heterogenosti uporabljenih omrežnih protokolov je lahko ta naloga zahtevne narave.

Kandidat naj v diplomski nalogi predstavi izbrane obstoječe rešitve za vizualizacijo večjih omrežij in naj razvije tudi svojo rešitev ter jo ovrednoti v perspektivi obstoječih. Implementacija lastne rešitve naj temelji na pridobivanju informacij o napravah od različnih protokolov, kot so SNMP, CDP in OSPF.

Mentor:

doc. dr. Zoran Bosnić

Dekan:

prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Matej Rojko,

z vpisno številko 63080182,

sem avtor diplomskega dela z naslovom:

Razvoj programskega paketa za izris omrežne topologije v velikih in srednje velikih podjetjih

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Zorana Bosnića,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identične s tiskano obliko diplomskega dela,
- soglašam z objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 1. september 2011

Podpis avtorja:

ZAHVALA

Za strokovno vodenje in usmerjanje se zahvaljujem svojemu mentorju, doc. dr. Zoranu Bosniću, še posebej pa se bi zahvalil svoji partnerki in staršem za podporo skozi študij.

Kazalo

1. Uvod	1
2. Rešitev za izris omrežne topologije.....	2
2.1 Protokoli, s pomočjo katerih deluje programski paket	2
2.1.1 Protokol SNMP	3
2.1.2 Protokol OSPF	5
2.1.3 Protokol CDP	5
2.2 Primerjave med programi	6
2.2.1 Programski paket IPSwitch	7
2.2.2 Programski paket SpiceWorks	8
2.2.3 Programski paket OpenNMS	9
2.2.4 Programski paket NetDisco.....	10
3. Razvoj programskega paketa za izris omrežne topologije	11
3.1 Opis postopka izdelave programskega paketa za izris omrežne topologije.....	11
3.1.1 Analiza omrežja in predvidevanje problemov	12
3.1.2 Konceptualni model	12
3.1.3 Logični model	14
3.2 Normalizacija podatkovnega modela.....	16
3.2.1 Določitev funkcionalnih odvisnosti	17
3.2.2 Prva normalna oblika	18
3.2.3 Druga normalna oblika.....	18
3.2.4 Tretja normalna oblika	19
3.3 Predstavitev programa	19
3.3.1 Zagon sistema.....	21
3.3.2 Vnos podatkov.....	21
3.3.3 Pregled topologije omrežja.....	22
3.3.4 Pregled omrežnih naprav v tabelah	22
3.4 Predstavitev implementacije programskega paketa za izris omrežne topologije.....	22
3.4.1 Implementacija uporabniškega vmesnika	22
3.4.2 Implementacija funkcije za pregled omrežja	22

3.4.3 Implementacija funkcije za izris omrežne topologije na komponento MS Visio	25
3.5 Predstavitev programskega paketa za izris omrežne topologije.....	26
3.5.1 Opis programskega paketa za izris omrežne topologije — pregledovalnik omrežja	28
3.5.2 Izris omrežne topologije	29
3.5.3 Pregled vmesnikov posameznih naprav	31
4. Sklepne ugotovitve.....	35
5. Kazalo slik.....	37
Literatura:.....	39

Kratice

SNMP – Simple Network Management Protocol

CDP – Cisco Discovery Protocol

OSPF – Open Shortest Path First

ARP – Address Resolution Protocol

WAN – Wide Area Network

IETF – Internet Engineering Task Force

MIB – Management Information Base

SMI – Structure of Management Information

NMA – Network Management Agents

MPLS – Multiprotocol Label Switching

URL – Uniform Resource Locators

ICMP – Internet Control Message Protocol

LLDP – Link Layer Discovery Protocol

EIGRP – Enhanced Interior Gateway Routing Protocol

RIP – Routing Information Protocol

BGP – Border Gateway Protocol

ATM – Asynchronous Transfer Mode

Povzetek

Glavni cilj diplomske naloge je razvoj programskega paketa za izris omrežne topologije večjih podjetjih in s tem olajšati omrežnemu administratorju skrb za dokumentacijo sistema. Diplomsko delo je razdeljeno na tri večja poglavja. V prvem poglavju smo opisali delovanje najpomembnejših protokolov, ki smo jih uporabili pri implementaciji programskega paketa.

Ti protokoli so: SNMP (angl. *Simple Network Management Protocol*), CDP (angl. *Cisco Discovery Protocol*), OSPF (angl. *Open Shortest Path First*).

Za opisom protokolov sledi primerjava programskega paketa z že obstoječimi rešitvami. Tu je opisano predvsem delovanje ostalih programskih rešitev, njihov način pridobivanja informacij iz omrežja in postopek namestitve.

V zadnjem delu diplomske naloge smo opisali glavne dele programskega paketa. Ti se nanašajo predvsem na izdelavo in normalizacijo podatkovne baze, prikaz diagrama primerov uporabe in opis implementacije nekaterih delov programskega paketa.

Abstract

The main objective of the thesis is the development of a new software package for plotting of the network topology of large companies and thus help the network administrator to make the system documentation. The thesis is divided into three major sections. The first chapter describes the main protocols which we used in the implementation of the software package.

These protocols are: SNMP (Simple Network Management Protocol), CDP (Cisco Discovery Protocol), OSPF (Open Shortest Path First).

After the description of the protocols, a comparison of the software packages with existing solutions follows. Here mostly the operations of other software solutions are described, their way of obtaining information from the network and the installation process.

In the last part of the thesis we described the major parts of our software package. These relate mostly to construction and normalisation of the database, the display of the diagram of cases used and the description of how we developed some parts of the software package.

1. Uvod

Vsa velika podjetja imajo svojo omrežno infrastrukturo, ki povprečno obsega nekaj sto omrežnih naprav. Za omrežno infrastrukturo štejemo usmerjevalnike in stikala, ki so med seboj povezani na fizičnem in logičnem nivoju. Poleg potreb po dodatni omrežni opremi je povečanje števila omrežnih naprav lahko tudi posledica sistematizacije namena aktivne opreme, kot določa v svojih priporočilih proizvajalec Cisco. Ta svojim strankam priporoča razdelitev omrežnih naprav na hrbtnični, porazdelitveni in dostopovni nivo. Z upoštevanjem tega zagotavljamo redundanco, s tem pa dobro stabilnost in odzivnost omrežij. Upravljanje s tako kompleksnimi omrežji je glavna skrb omrežnih administratorjev, ki za svoje učinkovito delo potrebujejo dobro dokumentiran sistem.

Izpad sistema pomeni za podjetja prekinitev poslovanja, zato morajo sistemski in omrežni administratorji imeti podroben pregled nad svojimi napravami. To nas je spodbudilo k razvoju programskega paketa za izris omrežne topologije. Slednji se izkaže za učinkovitega takrat, ko pride do težav, saj je za odpravljanje le-teh potrebno imeti dobro dokumentirano omrežje. Omrežna topologija mora vsebovati osvežene podatke, zato si omrežni administrator ne more privoščiti ročnega dodajanja in odstranjevanja podatkov v/iz dokumenta.

Namen diplomskega dela je podrobno predstaviti delovanje in učinkovitost programskega paketa ter primerjanje slednjega z obstoječimi rešitvami.

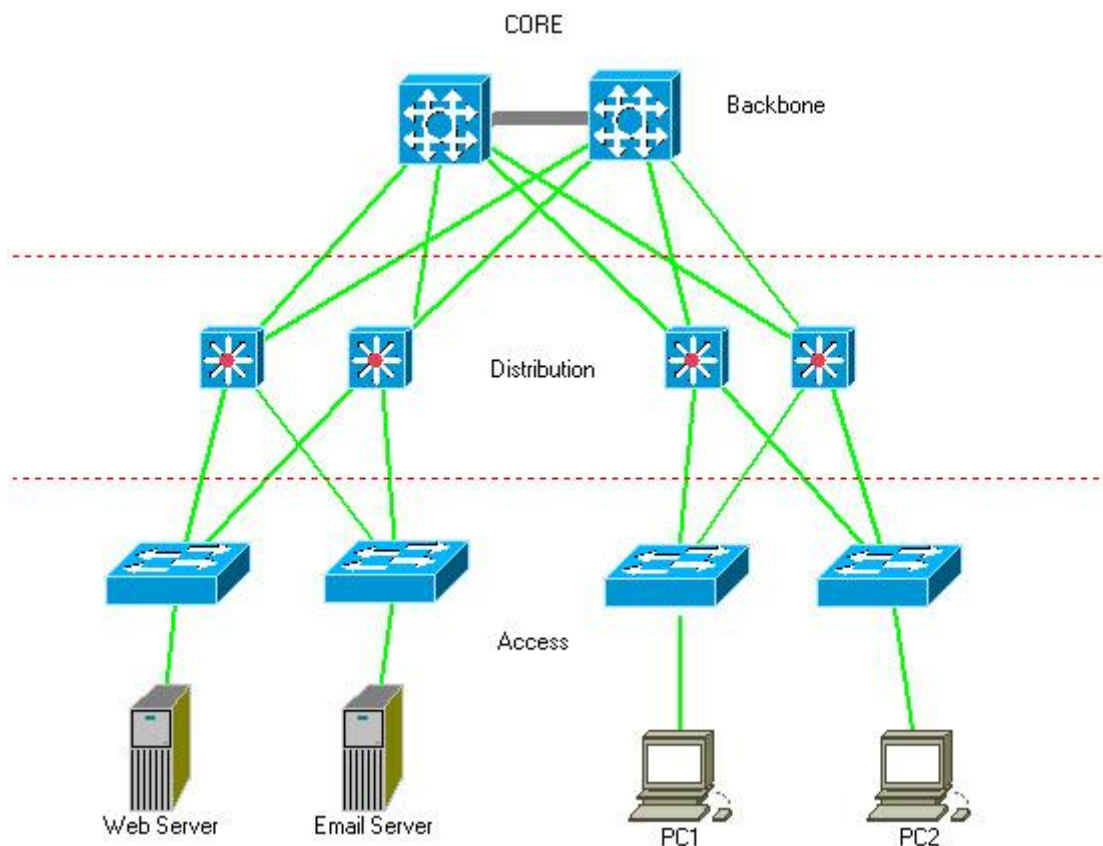
V prvem delu diplomske naloge bomo predstavili protokole, ki smo jih uporabili pri implementaciji programskega paketa. Tu bomo nekaj besed namenili protokolom SNMP (angl. *Simple Network Management Protocol*), OSPF (angl. *Open Shortest Path First*) in CDP (angl. *Cisco Discovery Protocol*). V sklopu prvega dela bo nekaj besed namenjenih primerjavi programskega paketa za izris omrežne infrastrukture z že obstoječimi rešitvami. Te dosejajo dobre rezultate pri nadzoru naprav v realnem času, žal pa nobena rešitev ni usmerjena v izris omrežne topologije v velikih podjetjih, kjer so podružnice med seboj povezane preko oblaka MPLS (angl. *Multiprotocol Label Switching*).

V drugem delu diplomske naloge se bomo osredotočili na postopke, ki smo jih uporabili pri implementaciji programskega paketa. Slednji se nanašajo na predstavitev podatkovne baze, ki daje glavno oporo našemu programskemu paketu. Tu bomo predstavili konceptualni in logični model podatkovne baze in njeno normalizacijo ter proceduro, ki preprečuje podvajanje podatkov pri vnosu v podatkovno bazo. Prikaz uporabe programskega paketa za izris omrežne topologije bomo predstavili s pomočjo diagrama primerov uporabe. Na koncu drugega dela bomo predstavili grafični vmesnik in opisali postopke in knjižnice, ki smo jih uporabili pri implementaciji.

2. Rešitev za izris omrežne topologije

2.1 Protokoli, s pomočjo katerih deluje programski paket

Vse naprave, ki jih človeštvo s pridom izkorišča za svoj dobrobit, vsebujejo veliko število komponent, ki jih moramo nadzorovati. Nadzor in skrb za stabilnost sta zelo pomembni za učinkovito delovanje sistema, s tem pa tudi celotnega podjetja. Vsa podjetja za svoje uspešno delovanje potrebujejo omrežno infrastrukturo, ki skrbi za delovanje računalnikov, strežnikov in drugih naprav v podjetju. Brez tega si sodobnih podjetij ne moremo predstavljati, zato je zelo pomembno, da sistem v podjetju deluje učinkovito in zanesljivo. Da bi kar največ časa v letu zagotavljali stabilnost sistema, moramo omrežno infrastrukturo dobro načrtovati in nadzorovati. Postopek načrtovanja traja dlje časa, seveda pa je dobro upoštevati priporočila proizvajalcev omrežnih naprav, kot je npr. Cisco, saj imajo na svojem področju dolgoletne izkušnje. Eno od priporočil slednjega proizvajalca je, da svojim strankam priporoča delitev omrežja na tri nivoje. Ti so prikazani na Slika 1.



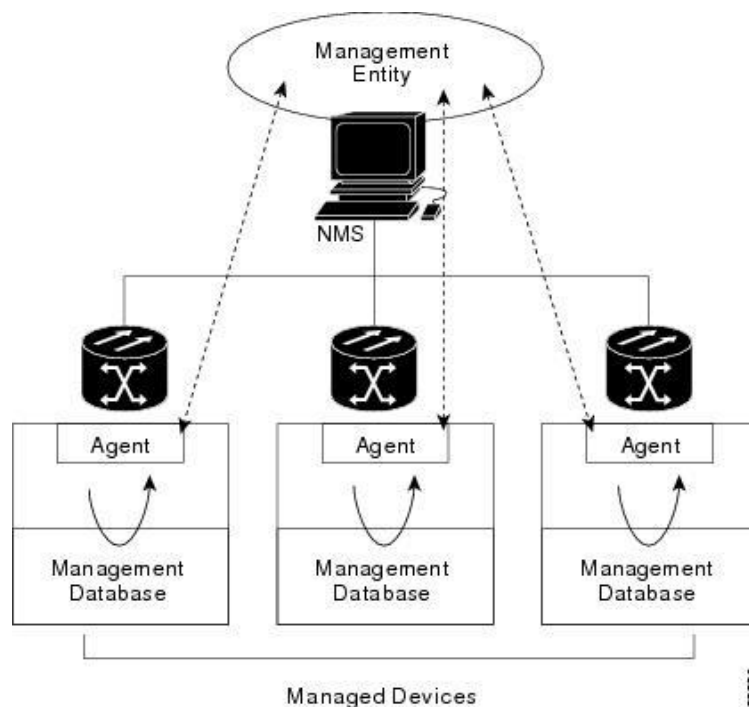
Slika 1: Prikaz razporeditve omrežnih naprav na tri nivoje [1].

Hrbtenični nivo (angl. *Core Layer*) vsebuje najmočnejše usmerjevalnike ali stikala, ki so med seboj povezani tako, da zagotavljajo 100 % redundanco med hrbteničnim in porazdelitvenim nivojem. Povezave med omrežnimi napravami v tem delu dosega do 10 GB/s, stabilnost ostalega dela omrežja pa je odvisna predvsem od teh naprav. Pod hrbteničnim sledi porazdelitveni nivo, ki skrbi za WAN (angl. *Wide Area Network*) povezave med različnimi podružnicami v podjetju. Ta del vsebuje naprave, ki so zmožne preklapljati in usmerjati promet. Slednje delujejo med drugim in tretjim nivojem modela ISO/OSI. Nivo, ki loči uporabnike od porazdelitvenega omrežja, se imenuje dostopni nivo.

Že minimalna hierarhija, ki sledi zgornjemu priporočilu, vsebuje kar nekaj omrežnih naprav, seveda pa se z velikostjo podjetja število in potrebe še povečajo. Skrb za taka omrežja je kompleksna in prav zato so strokovnjaki iz organizacije IETF (angl. *Internet Engineering Task Force*) definirali protokol SNMP.

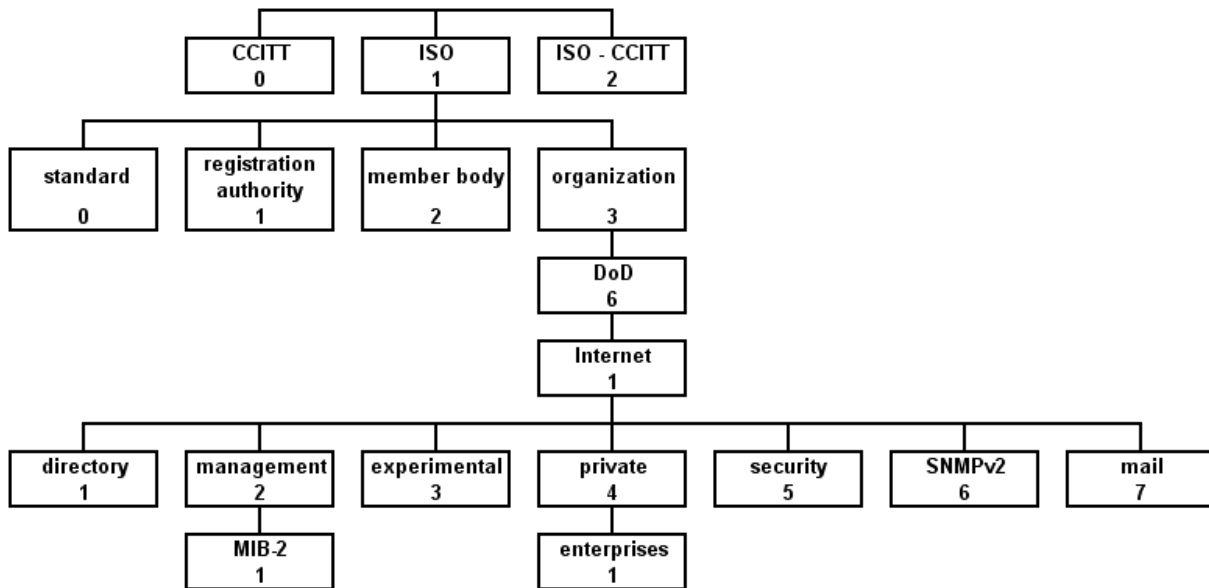
2.1.1 Protokol SNMP

Protokol SNMP je jezik, s pomočjo katerega si lahko agent in strežnik med seboj izmenjujeta sporočila. Vsaka omrežna naprava, ki ima omogočeno funkcijo za branje informacij preko protokola SNMP, mora imeti nameščenega agenta NMA (angl. *Network Management Agents*). Ta nadzornemu sistemu daje možnost branja podatkov na zahtevo, kar pomeni, da se mora agent odzvati in ponuditi nadzornemu sistemu zahtevane podatke in te predstaviti v naprej definirani obliki. Slika 2 prikazuje agente in upravljavca omrežja.



Slika 2: Prikaz odnosa agenta in upravljavca omrežja [2].

Zaradi možnosti nadgradnje posameznega protokola so morali pri organizaciji IETF določiti, na kakšen način bodo informacije o napravah predstavljene. Slednje določa MIB (angl. Management Information Base) vrednost, ki predstavlja hierarhično obliko porazdeljenih objektov in parametrov z informacijami o napravi, kjer se na posamezne objekte ali parametre sklicujemo s hierarhično številko, ki jo določa drevo na Slika 3. MIB vrednost je lahko na vsaki napravi različna in je predvsem odvisna od proizvajalcev naprav. Tu moramo omeniti, da so nekatere znane vrednosti (npr. imena naprav), zapisane z istimi MIB vrednostmi na vseh omrežnih napravah.



Slika 3: Hierarhično drevo ASN.1 za naslavljanje komponent v MIB [3].

Pri organizaciji IETF so se odločili, da do vrednosti objekta dostopamo s pomočjo drevesa, ki je prikazano na Slika 3. Do želene vrednosti pridemo tako, da se skozi drevo pomikamo do lista, katerega vrednost nas zanima [4].

Omrežna infrastruktura lahko vsebuje naprave različnih proizvajalcev. Če želimo s temi napravami v omrežju enotno in učinkovito upravljati, potrebujemo konsistenten in standardiziran opis podatkov. Jezik, ki definira te podatke, se imenuje SMI (angl. *Structure of Management Information*). S pomočjo SMI definiramo podatkovne tipe objektov, ki se nahajajo na napravah in do katerih dostopamo s pomočjo MIB zapisa. SMI definira enajst različnih podatkovnih tipov, ti so:

- INTEGER,
- Integer32,
- Unsigned32,
- OCTET STRING,

- OBJECT IDENTIFIER,
- IPAddress,
- Counter32,
- Counter64,
- Gauge32,
- TimeTicks,
- Opaque.

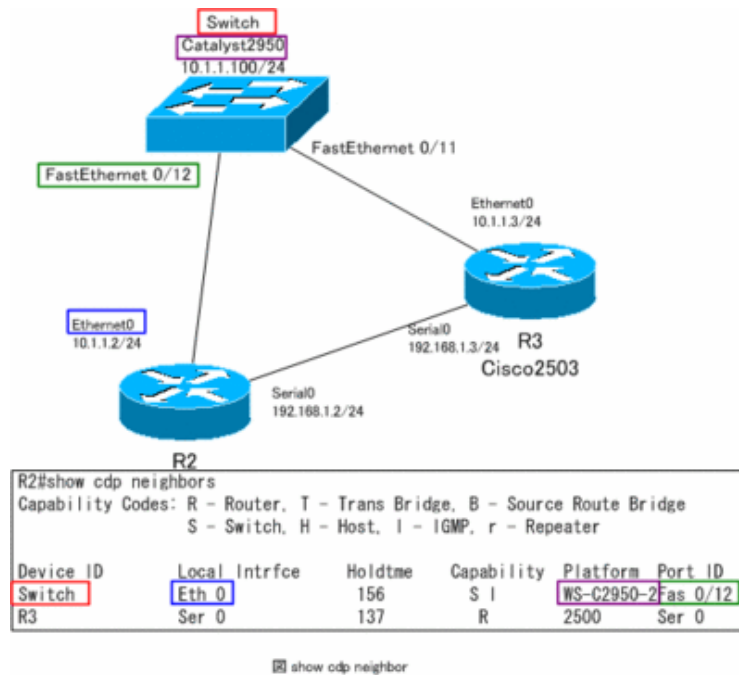
Več o podatkovnih tipih lahko najdete v viru [5].

2.1.2 Protokol OSPF

Na začetku načrtovanja programskega paketa za izris omrežne topologije smo se odločili za simuliranje protokola OSPF. Slednji deluje na tretjem nivoju modela ISO/OSI, kar z drugimi besedami pomeni, da je protokol namenjen za usmerjanje paketov med različnimi usmerjevalniki. Kot je navedeno v viru [6], protokol OSPF uporablja usmerjanje na podlagi stanja povezav (ang. *Link-State*), tako da si vsak usmerjevalnik v internem omrežju zgradi topologijo omrežja in s pomočjo algoritma Dijkstra določi najugodnejše poti do ostalih usmerjevalnikov. Izbira poti je odvisna od cene poti, ki vključuje hitrosti povezav do sosednjih usmerjevalnikov.

2.1.3 Protokol CDP

Za popolno sliko omrežne topologije smo morali pregledati tako usmerjevalnike kot stikala. Pregled stikal smo opravili s pomočjo simuliranja protokola CDP, ki deluje na drugi plasti modela ISO/OSI. Protokol CDP je razvilo podjetje Cisco z namenom, da si sosednje naprave v omrežju izmenjujejo informacije. Protokol CDP deluje na način, kot je to opisano v viru [7] (tj. tako, da vsaka naprava pošlje sosednji napravi informacije o sebi). V primeru, da sosednja naprava, ki prejme paket, podpira protokol CDP, zgradi tabelo, ki jo lahko preberemo s pomočjo ukaza: »*show cdp neighbors*«. Na Slika 4 vidimo izpis tabele, ki hrani podatke o svojih sosednjih napravah.



Slika 4: Prikaz izpisa ukaza show cdp neighbors [8].

2.2 Primerjave med programi

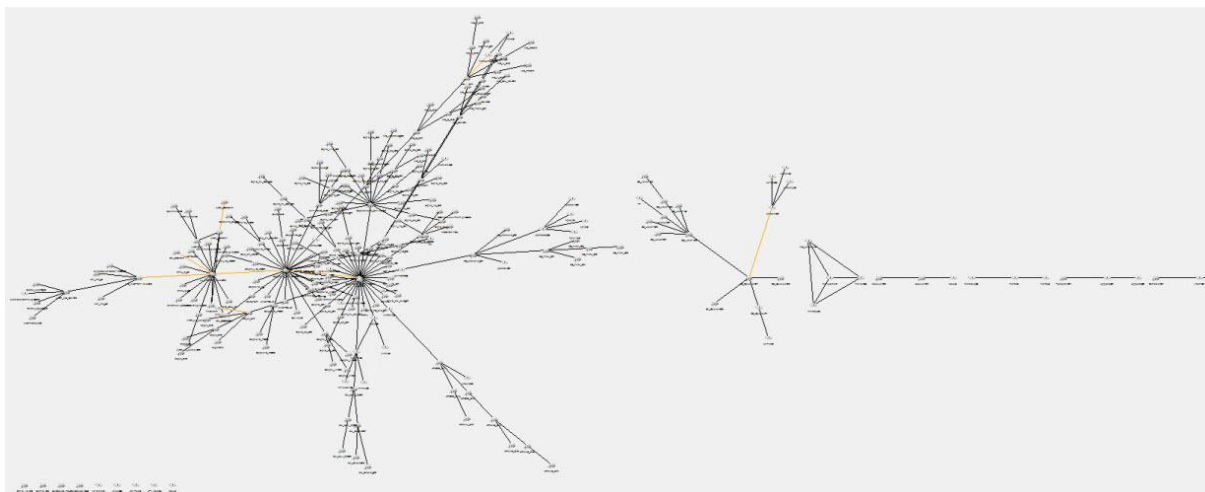
V času načrtovanja novega programskega paketa za izris omrežne topologije smo preverili, kateri produkti na to temo že obstajajo. Slednjih je kar nekaj, vendar je večina izmed njih namenjena nadzoru omrežja, nobeden od njih pa ni ozko usmerjen v izris omrežne topologije, ki v večjem okolju večini programskih paketov povzroča težave. Ob pregledu obstoječih programskih paketov smo preverili tako odprtokodne kot tudi plačljive rešitve. Za slednje moramo imeti na računalniku nameščen operacijski sistem Windows, saj so navadno razviti v razvojnem okolju Microsoft Visual Studio.

Pri odprtokodnih rešitvah je nekoliko drugače. Slednji delujejo na način strežnik – odjemalec, kjer je strežnik predstavljen s pomočjo virtualnega računalnika, na katerem je nameščen operacijski sistem Linux, odjemalec pa dostopa do podatkov s pomočjo internetnega brskalnika. V tem primeru je za uporabnika vseeno, kateri operacijski sistem ima nameščen na računalniku. Ta programska rešitev deluje tako, da se odjemalec poveže na naslov, na katerem posluša strežnik. Ko je zahteva podana, odjemalec s pomočjo internetnega brskalnika prikaže rezultate. Preizkusili smo naslednje programske pakete:

- IPSwitch,
- SpiceWorks,
- NetDisco,
- OpenNMS.

2.2.1 Programski paket IPSwitch

Programski paket IPSwitch je plačljiv programski paket podjetja WhatsUpGold. Izris omrežne topologije večjega telekomunikacijskega podjetja je bil s pomočjo tega programa še najbližji približek zelenemu rezultatu. Programski paket deluje na način, da na vsaki napravi preveri tabelo ARP (angl. *Address Resolution Protocol*) in na podlagi pridobljenih podatkov izriše omrežno topologijo. Pregled omrežja je trajal približno 30 minut, rezultat pa je prikazan na Slika 5.



Slika 5: Prikaz izrisa omrežne topologije s pomočjo programskega paketa IPSwitch.

S Slika 5 je razvidno, da so nekateri deli omrežja nepovezani. Glavno težavo v tem primeru predstavlja oblak MPLS. Tega si lahko predstavljamo kot oblak, v katerem so usmerjevalniki, ki niso zanimivi za našo interno omrežno topologijo. Skozi oblak MPLS se med dvema usmerjevalnikoma vzpostavi navidezni vod, ki ga pri pregledovanju omrežja ne smemo zanemariti. Več o načinu delovanja MPLS omrežja je moč najti v viru [9]. Nepravilno sliko omrežja lahko dobimo, kadar ne znamo pravilno povezati naprav, ki so med seboj na logičnem nivoju povezane skozi oblak MPLS ali ATM (angl. *Asynchronous Transfer Mode*). Te težave se pojavijo, ker iz tabele ARP ne moremo pravilno razbrati, katera naprava je onkraj oblaka MPLS soseda trenutni napravi, na kateri je bila sprožena zahteva ARP. Poleg omenjenih težav moramo biti pri pregledu omrežja pazljivi na podvajanje kombinacij sosednjih naprav. Ker se ena naprava lahko dvema različnima napravama predstavi z različnim naslovom IP, lahko pride pri izrisu omrežne topologije do podvajanja naprav.

Druga večja težava pri uporabi protokola ARP nastane, kadar je naprava dlje časa neaktivna. To ne pomeni, da je naprava izključena iz električnega omrežja, temveč da na napravo niso priključeni nobeni končni uporabniki, ki bi ustvarjali promet. Taka naprava bo zaradi preteka časovnega intervala za hranjenje v tabeli ARP iz nje izbrisana.

2.2.2 Programski paket SpiceWorks

Programski paket SpiceWorks deluje na način odjemalec – strežnik. Strežnik namestimo na računalnik, na katerem mora biti nameščen operacijski sistem Windows. Po končani namestitvi odpremo internetni brskalnik in v polje URL (angl. *Uniform Resource Locators*) vnesemo naslov, na katerem posluša strežniški del programskega paketa. Uporabniški vmesnik je zelo prijazen in pregleden za uporabnika, vendar se pojavijo težave, kadar želimo programski paket uporabiti v večjih okoljih. Ko smo želeli izrisati omrežno topologijo večjega komunikacijskega podjetja, program te naloge ni uspešno opravil, saj v programu ne moremo določiti, katere naprave bi radi izrisali in katerih ne. SpiceWorks je pri pregledu celotnega omrežja zajel omrežne naprave, kot so usmerjevalniki in stikala, poleg njih pa še vse računalnike, strežnike in IP telefone. Ker je bilo na tak način najdenih 3000 naprav, vmesnik za izris omrežne topologije enostavno ni zmožal opraviti svoje naloge.

Prikaz nastavitvev za pregled omrežja prikazuje Slika 6, s katere je razvidno, da lahko uporabnik določi več omrežnih naslovov hkrati in nastavi sprožilec začetka izvajanja pregleda omrežja.

The screenshot displays the 'Settings' window for 'Network Scan'. At the top, there are buttons for 'New Scan Entry' and 'Start Network Scan'. Below this is the 'Scan Entries' section, which contains a table with three entries:

ID	IP Range	Protocol	Status
1	172.22.1.1-255	TSLmrojk (Windows) Public (SNMP)	Enabled
2	192.168.119.1-255	TSLmrojk (Windows) Public (SNMP)	Enabled
3	172.17.17.1-254	TSLmrojk (Windows) Public (SNMP)	Disabled

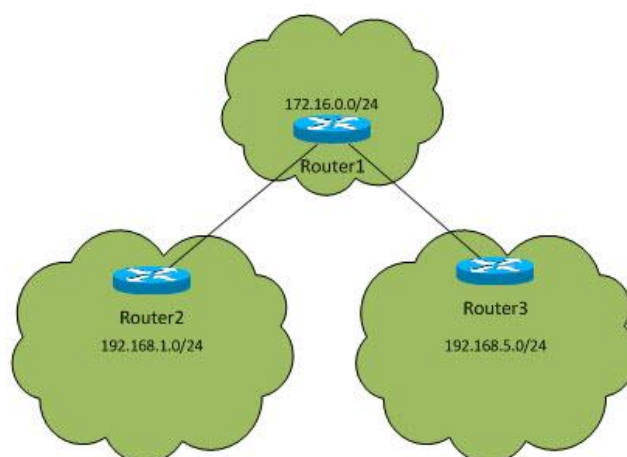
Each entry has a 'Scan Schedule: default schedule' and a 'Click here' link. Below the entries is the 'Scan Exclusions' section with a text input field and a 'Save' button. The 'Default Schedule' section includes a 'Run on:' section with checkboxes for days of the week (all checked), a 'Start at:' dropdown set to '4 AM', and a 'Repeat every:' dropdown set to '12 hours'. There are 'Save' and 'Cancel' buttons. The 'Global Scan Settings' section includes several dropdown menus: 'Scheduled scan:' (Enabled), 'Scan speed:' (Medium), 'Incremental scanning:' (Enabled), 'Rescan interval:' (18 hours), and 'Wake-On LAN:' (Disabled). Each dropdown has a small explanatory text to its right. 'Save' and 'Cancel' buttons are at the bottom.

Slika 6: Prikaz definiranja območja, ki naj ga program pregleda.

V primeru, ko mora omrežni administrator pregledati večje omrežje, mu lahko način, ki je prikazan na Slika 6, povzroča težave, saj smo z našo analizo programa ugotovili, da je najbolje postopoma pregledovati omrežje in sprti klasificirati neznane naprave v ustrezne skupine. V primeru, da bi uporabnik določil, naj programski paket pregleda vse naslove od 0.0.0.0 do 255.255.255.255, bi funkcija za pregled našla veliko naprav, ki ne bi bile razporejene v skupine usmerjevalnikov, stikal, telefonov IP ali tiskalnikov ampak bi slednje naprave bile dane v skupino, ki se imenuje neznane naprave. Ta skupina nam bi pri nadaljnjem razvrščanju naprav povzročala težave. Najbolj primeren pregledovanja omrežja je prikazan v naslednjem primeru.

Postopek za pregled omrežja, ki je prikazano na Slika 7:

- uporabnik mora najprej pregledati omrežje 172.16.0.0/24,
- po končanem pregledu je potrebno klasificirati vse neznane naprave v ustrezne skupine,
- nato preverimo, katera nova omrežja so bila najdena (npr. 192.168.1.0 in 192.168.5.0),
- za vsa na novo najdena omrežja ponovimo postopek.



Slika 7: Prikaz različnih omrežij.

2.2.3 Programski paket OpenNMS

Programski paket OpenNMS je najboljša izbira za nadzor omrežja, saj sta v njem zelo dobro rešena tako pregled kot zbiranje informacij o napravah. Nadzor se nanaša na:

- pretok prometa skozi omrežno napravo,
- delovanje naprave,

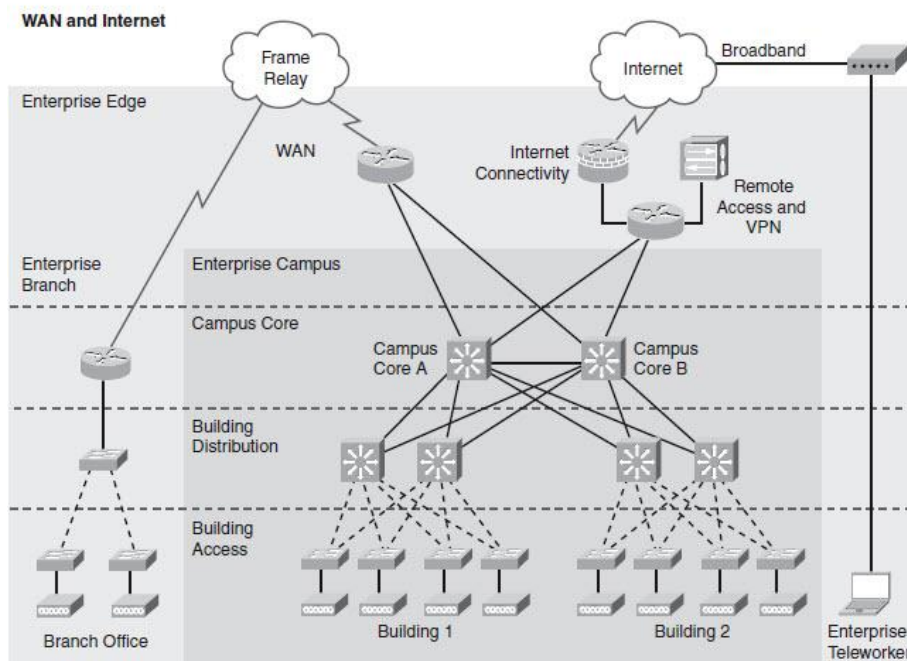
- prikaz opozoril.

Programski paket pri pregledu omrežij deluje na podoben način kot programski paket SpiceWorks, torej omrežni administrator določi območje, v katerem bo program iskal delujoče naprave, s to razliko, da lahko uporabnik programskega paketa OpenNMS določi še, katere tipe naprav želi iskati.

Oba programska paketa iščeta delujoče naprave s pomočjo protokola ICMP (angl. *Internet Control Message Protocol*). Pregledovanje omrežja deluje tako, da program na vsak naslov v območju naslovov, ki jih je določil uporabnik, pošlje paket ICMP in posluša, če je ta naslov dodeljen napravi v omrežju. V primeru, da je naslov dodeljen napravi, se bo ta tudi odzvala. Program nato s pomočjo protokola SNMP pridobi informacije vseh naprav, kjer je bil odziv na paket ICMP uspešen. Žal je glavna težava tega programskega paketa ta, da ni namenjen izrisu omrežne topologije, zelo dobro pa vodi informacije o posameznih napravah v omrežju.

2.2.4 Programski paket NetDisco

NetDisco je odprtokodna programska rešitev za upravljanje z omrežnimi napravami v večjih korporacijah. Slednji je zasnovan na način odjemalec – strežnik. Za svoje delovanje uporablja protokola CDP in LLDP (angl. *Link Layer Discovery Protocol*). Oba protokola delujeta na drugem nivoju modela ISO/OSI. Problem programa se pojavi, če ga želimo uporabljati v podjetju, ki ima več podružnic, ki so med seboj povezane preko oblaka MPLS, ATM ali Frame Relay, kot kaže Slika 8.



Slika 8: Prikaz načina povezave v večjem podjetju.

S Slika 8 je moč ugotoviti, da je v večjih podjetjih poleg drugega prisoten še tretji nivo modela ISO/OSI. To pomeni, da moramo pri pregledu omrežij upoštevati tudi usmerjevalne protokole, kot so:

- OSPF,
- EIGRP (angl. *Enhanced Interior Gateway Routing Protocol*),
- RIP (angl. *Routing Information Protocol*),
- BGP (angl. *Border Gateway Protocol*).

Kot smo že omenili, programski paket za pregled omrežja uporablja le protokola CDP ali LLDP, s katerima pa ni moč zaznati vseh omrežnih naprav v podjetju. Število naprav, ki jih ne zaznamo, je lahko kar veliko, saj so le te lahko povezane z usmerjevalnikom, ki ga programski paket pri pregledovanju omrežja na drugem nivoju modela ISO/OSI ne zazna. Kljub temu nam je bil ta program pri razvoju našega programskega paketa v veliko pomoč, saj smo tako prišli do ugotovitve, da je potrebno za doseg želenega cilja poleg protokola CDP uporabiti še protokole, ki delujejo na tretjem nivoju modela ISO/OSI.

Programski paket NetDisco je potrebno namestiti na odprtokodni operacijski sistem Linux, za katerega obstajajo dobre procedure, ki uporabniku olajšajo delo pri namestitvi vseh potrebnih knjižnic. Ko je namestitev programa uspešno zaključena, mora uporabnik preko strežnika omogočiti ali onemogočiti funkcije, ki mu jih nudi programski paket NetDisco. Skozi obdobje testiranja smo ugotovili, da je slednji namenjen predvsem nadzoru omrežnih naprav, tako kot programski paket OpenNMS. Žal smo bili tudi tu razočarani s sliko topologije omrežja, saj programu ni uspelo povezati vseh naprav preko omrežja MPLS.

3. Razvoj programskega paketa za izris omrežne topologije

3.1 Opis postopka izdelave programskega paketa za izris omrežne topologije

Postopek izdelave programskega paketa je zahteval rešitev naslednjih nalog:

- analiza omrežja in predvidevanje problemov,
- načrtovanje podatkovne baze,
- izdelava grafičnega vmesnika,
- rešitev problema rekurzivnega pregledovanja omrežnih naprav,

- rešitev problema redundantnih povezav,
- rešitev problema zaznavanja oblaka MPLS,
- rešitev problema izrisa omrežnih naprav na komponento Microsoft Visio.

3.1.1 Analiza omrežja in predvidevanje problemov

Programski paket za izris omrežne topologije smo razvijali v večjem komunikacijskem podjetju, zato smo morali na začetku pridobiti informacije o:

- številu omrežnih naprav (npr. usmerjevalnikov in stikal),
- usmerjevalnih protokolih, ki jih usmerjevalniki uporabljajo za sporazumevanje,
- približno število oblakov MPLS,
- območju omrežja, kjer ne želimo pridobiti nobenih informacij,
- verziji, ki jo uporablja protokol SNMP.

Kljub dobremu premisleku o problemih, ki bi lahko nastali skozi obdobje razvoja programskega paketa, smo se zavedali, da vseh težav na začetku ne bomo mogli predvideti, zato smo se odločili, da bomo podatkovno bazo načrtovali tako, da bo v poznejših obdobjih mogoče popraviti katerikoli njen del. To smo dosegli z metodologijo razvoja, ki se imenuje »po delih«. Ta metodologija omogoča, da najprej razvijemo okvirno shemo z najpomembnejšimi entitetnimi tipi in razmerji, pozneje pa to shemo širimo, dokler je potrebno. Za izognitev anomalijam smo podatkovni model normalizirali do tretje normalne oblike in se s tem izognili parcialnim in tranzitivnim odvisnostim. Načrtovanje podatkovnega modela vsebuje tri različne postopke:

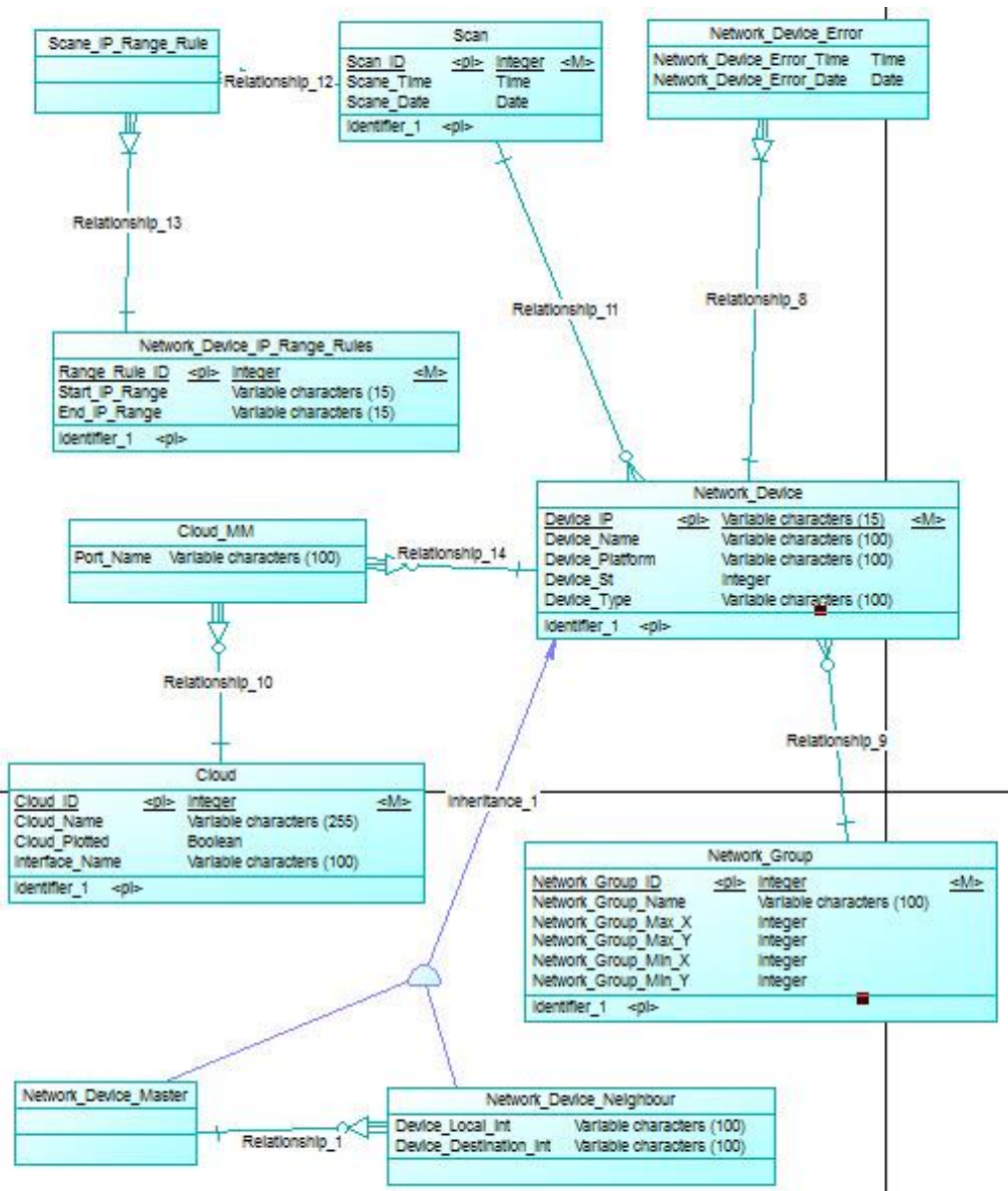
- načrtovanje konceptualnega modela,
- pretvorba v logični model,
- pretvorba v fizični model.

3.1.2 Konceptualni model

Načrtovanje podatkovne baze smo pričeli z načrtovanjem konceptualnega modela [10]. Ta je po koncu načrtovanja vseboval deset tabel, ki so bile povezane med seboj z različnimi vrstami razmerij (npr. 1:n, n:m ...). Poleg vrste razmerij je s Slika 9 razvidno, da konceptualni model vsebuje tudi posplošitev entitet, kar pomeni, da tabela z naslovom »omrežna naprava« (Network Device) pri pretvarjanju konceptualnega modela v logični model posreduje attribute tabelama:

- glavna omrežna naprava (Network_Device_Master),
- sosednja omrežna naprava (Network_Device_Neighbor).

Ti tabeli imata v programskem paketu za izris omrežne topologije posebno funkcijo, saj se z njuno pomočjo uspešno izvede rekurzija, ki skrbi za pregled naprav v omrežju. Tabela z naslovom »glavna omrežna naprava« hrani podatke o napravah, ki imajo sosede, medtem ko tabela »sosednja omrežna naprava« vsebuje informacije o končnih napravah.



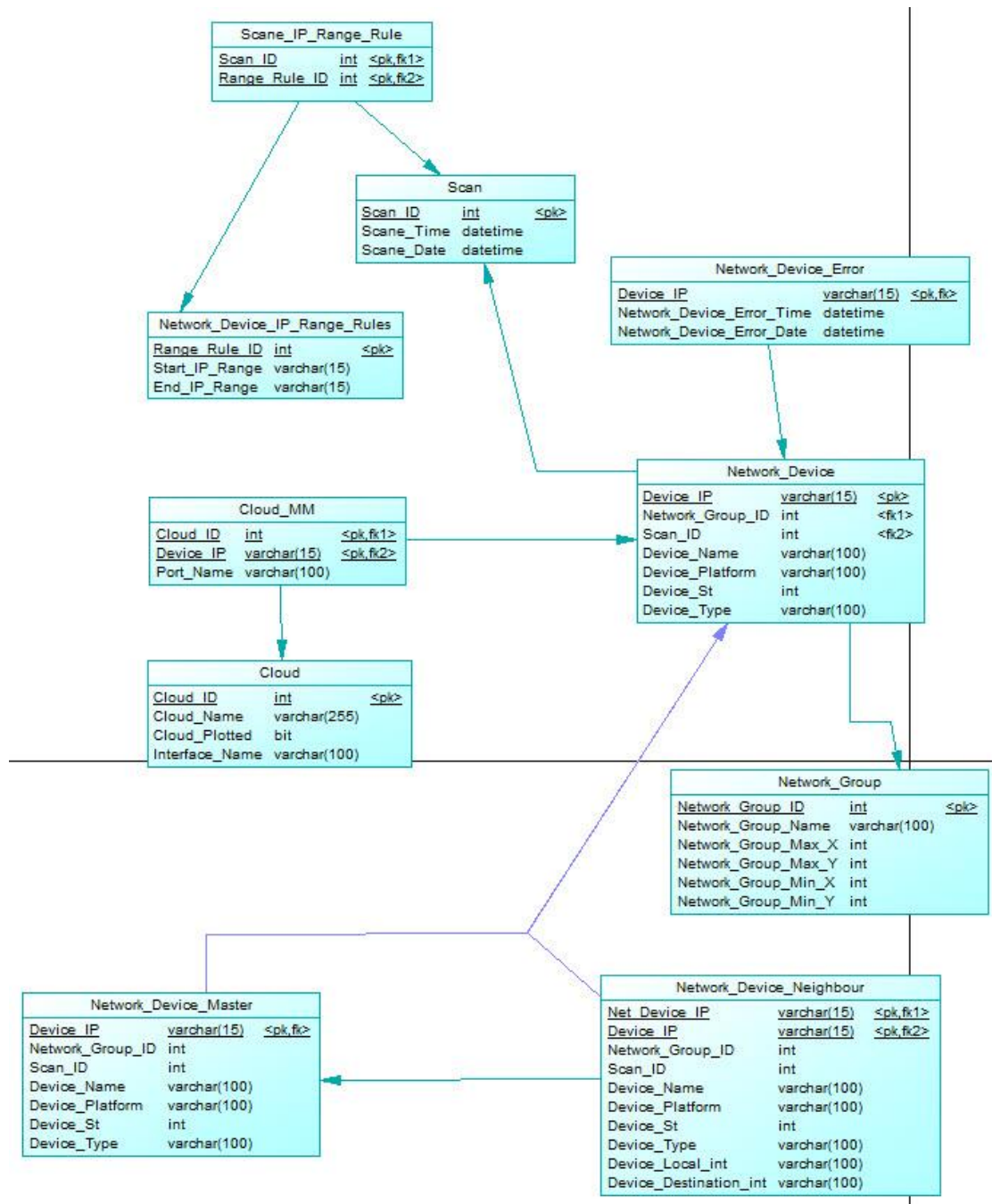
Slika 9: Prikaz konceptualnega modela podatkovne baze.

3.1.3 Logični model

Po končanem načrtovanju konceptualnega modela podatkovne baze smo le tega pretvorili v logični model. Pri pretvarjanju v logični modela se vsi entitetni tipi pretvorijo v:

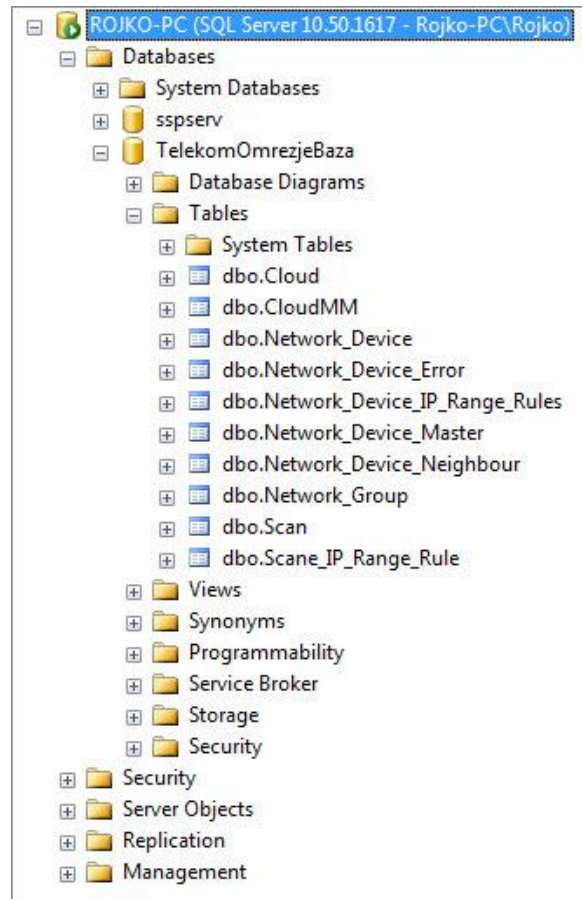
- šibke entitetne tipe (to pomeni, da je ključ sheme šibkega entitetnega tipa odvisen od primarnega ključa in ključev ostalih entitetnih tipov, s katerimi je povezan v razmerju n:1 ali n:0. Npr. entitetni tip `Network_Device_Neighbor`),
- močne entitetne tipe (to pomeni, da je ključ sheme močnega entitetnega tipa enolično določen v entitetnemu tipu. Npr. entitetni tip `Cloud`).

Poleg ključev se opravi tudi posplošitev med entitetnimi tipi. To pomeni, da se vsi atributi tabele »`Network_Device`« kopirajo v tabeli »`Network_Device_Neighbor`« in »`Network_Device_Master`«. Pretvorba je prikazana na Slika 10, kjer opazimo, da imajo vsi trije entitetni tipi enaka imena atributov. Posplošitev je zelo pomembna, saj je tako moč razbrati, da nimamo opisa treh popolnoma različnih naprav, vendar gre tu za enake naprave, ki v različnih okoljih potrebujejo različne opise.



Slika 10: Prikaz logičnega modela podatkovne baze.

Po končani pretvorbi iz konceptualnega v logični model moramo logični model pretvoriti še v fizičnega. Program za načrtovanje podatkovne baze (PowerDesigner) nam sam generira stavke SQL, ki jih pozneje uporabimo v poizvedbi programa Microsoft SQL Server 2008 R2. Po končani izvedbi poizvedbe se v programskem paketu Microsoft SQL Server 2008 R2 prikažejo tabele, tako kot nam to prikazuje Slika 11.



Slika 11: Prikaz tabel v programskem paketu Microsoft SQL Server 2008 R2.

Podatkovni model nam predstavlja osnovni del našega programskega paketa, saj smo s pomočjo njega uspeli hraniti podatke o napravah v omrežju. Veliko vlogo v programskem paketu predstavljajo tudi procedure, ki preverjajo, dodajajo, vračajo in posodablajo podatke v podatkovni bazi.

3.2 Normalizacija podatkovnega modela

Normalizacija podatkovnega modela ima pri načrtovanju programskega paketa pomembno vlogo, saj se z njeno pomočjo izognemo različnim anomalijam. Največkrat se podatkovni model normalizira do tretje normalne oblike, ki že odpravlja tranzitivne odvisnosti, to pa je dovolj za običajno uporabo podatkovnega modela. Koraki pri normalizaciji podatkovne baze so:

- določitev funkcionalnih odvisnosti,
- normalizacija v prvo normalno obliko,
- normalizacija v drugo normalno obliko,
- normalizacija v tretjo normalno obliko.

3.2.1 Določitev funkcionalnih odvisnosti

Def [11]: Če sta A in B atributa relacije R, potem je atribut B funkcionalno odvisen od atributa A (notacija: $A \rightarrow B$), če je vsaki vrednosti atributa A pridružena natanko ena vrednost atributa B. V tem primeru atribut A predstavlja determinanto za atribut B. Za funkcionalno odvisnost lahko rečemo, da je to lastnost pomena (semantike) atributov relacije.

Poznamo tri vrste funkcionalnih odvisnosti:

- delna funkcionalna odvisnost (kadar je atribut relacije funkcionalno odvisen le od dela ključa relacije),
- polna funkcionalna odvisnost (kadar je atribut funkcionalno odvisen od celotnega ključa),
- tranzitivna funkcionalna odvisnost (kadar je atribut posredno funkcionalno odvisen od ključa tabele).

S pomočjo Slika 10 lahko po zgornji definiciji razberemo naslednje funkcionalne odvisnosti:

Tabela (»Network_Device_IP_Range_Rules«): <u>Range_Rule_ID</u> -> Start_IP_Range; <u>Range_Rule_ID</u> -> End_IP_Range;	Tabela (»Scan«): <u>Scan_ID</u> -> Scan_Time; <u>Scan_ID</u> -> Scan_Date;
Tabela (»Network_Device_Error«): <u>Device_IP</u> -> Network_Device_Error_Time; <u>Device_IP</u> -> Network_Device_Error_Date;	Tabela (»Cloud_MM«): <u>Cloud_ID,Device_IP</u> -> Port_Name;
Tabela (»Cloud«): <u>Cloud_ID</u> -> Cloud_Name; <u>Cloud_ID</u> -> Clous_Plotted; <u>Cloud_ID</u> -> Interface_Name;	Tabela (»Network_Group«): <u>Network_Group_ID</u> -> Network_Group_Name; <u>Network_Group_ID</u> -> Network_Group_Max_X; <u>Network_Group_ID</u> -> Network_Group_Max_Y; <u>Network_Group_ID</u> -> Network_Group_Min_X; <u>Network_Group_ID</u> -> Network_Group_Min_Y;
Tabela (»Network_Device_Master«): <u>Device_IP</u> -> Network_Group_ID; <u>Device_IP</u> -> Scan_ID; <u>Device_IP</u> -> Device_Name; <u>Device_IP</u> -> Device_Platform; <u>Device_IP</u> -> Device_St; <u>Device_IP</u> -> Device_Type;	Tabela (»Network_Device_Neighbor«): M_D_IP = »Master_Device_IP«, N_D_IP = »Neighbor_Device_IP«, <u>M_D_IP,N_D_IP</u> ->Device_Local_Int; <u>M_D_IP,N_D_IP</u> ->Device__Destination_Int; <u>N_D_IP</u> ->Device_Name; <u>N_D_IP</u> ->Device_Platform; <u>N_D_IP</u> ->Device_St; <u>N_D_IP</u> ->Device_Type;
Tabela (»Network_Device«): <u>Device_IP</u> -> Device_Name; <u>Device_IP</u> -> Device_Platform;	

<u>Device_IP</u> -> Device_St;	
<u>Device_IP</u> -> Device_Type;	

3.2.2 Prva normalna oblika

Tabela je v prvi normalni obliki:

- kadar so določeni vsi ključni atributi,
- kadar ne vsebuje ponavljajočih se skupin vrednosti atributov.

Naš podatkovni model je v prvi normalni obliki, saj imajo vsi entitetni tipi določene primarne ključe in se v nobeni tabeli atributi ne ponavljajo [12].

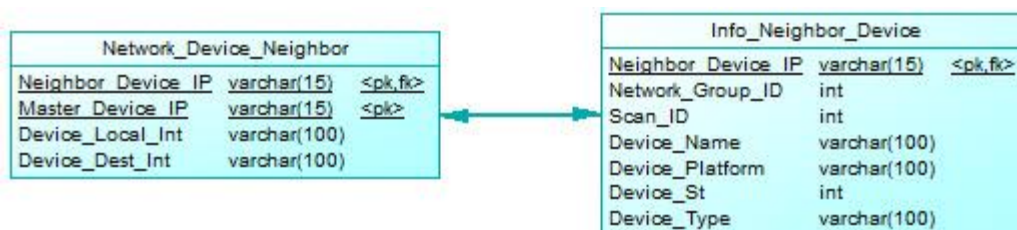
3.2.3 Druga normalna oblika

Relacija je v drugi normalni obliki, če se nahaja v prvi normalni obliki in je vsak neključni atribut polno funkcionalno odvisen od primarnega ključa (če ne obstaja nobena delna funkcionalna odvisnost) [13].

Postopek transformacije iz prve normalne oblike v drugo normalno obliko:

- ugotovimo primarni ključ relacije v prvi normalni obliki,
- opredelimo funkcionalne odvisnosti znotraj relacije,
- če obstajajo atributi, ki so delno funkcionalno odvisni od primarnega ključa, ga skupaj z njegovo determinanto prestavimo v novo relacijo. Determinanta ostane tudi v osnovni relaciji.

Vse tabele, ki nimajo sestavljenega ključa, so po definiciji v drugi normalni obliki. V našem primeru tabela »Network_Device_Neighbor« ni v drugi normalni obliki, saj so določeni atributi odvisni le od dela ključa. Rešitev za to tabelo je prikazana na Slika 12.



Slika 12: Prikaz pretvorbe tabele "Network_Device_Neighbor" v drugo normalno obliko.

S slike 12 je razvidno da smo ključ »Neighbor_Device_IP« prestavili v novo tabelo »Info_Neighbor_Device«, poleg tega smo prestavili še vse attribute, ki so bili odvisni le od primarnega ključa »Neighbor_Device_IP«. Tako smo naš podatkovni model pretvorili v drugo normalno obliko.

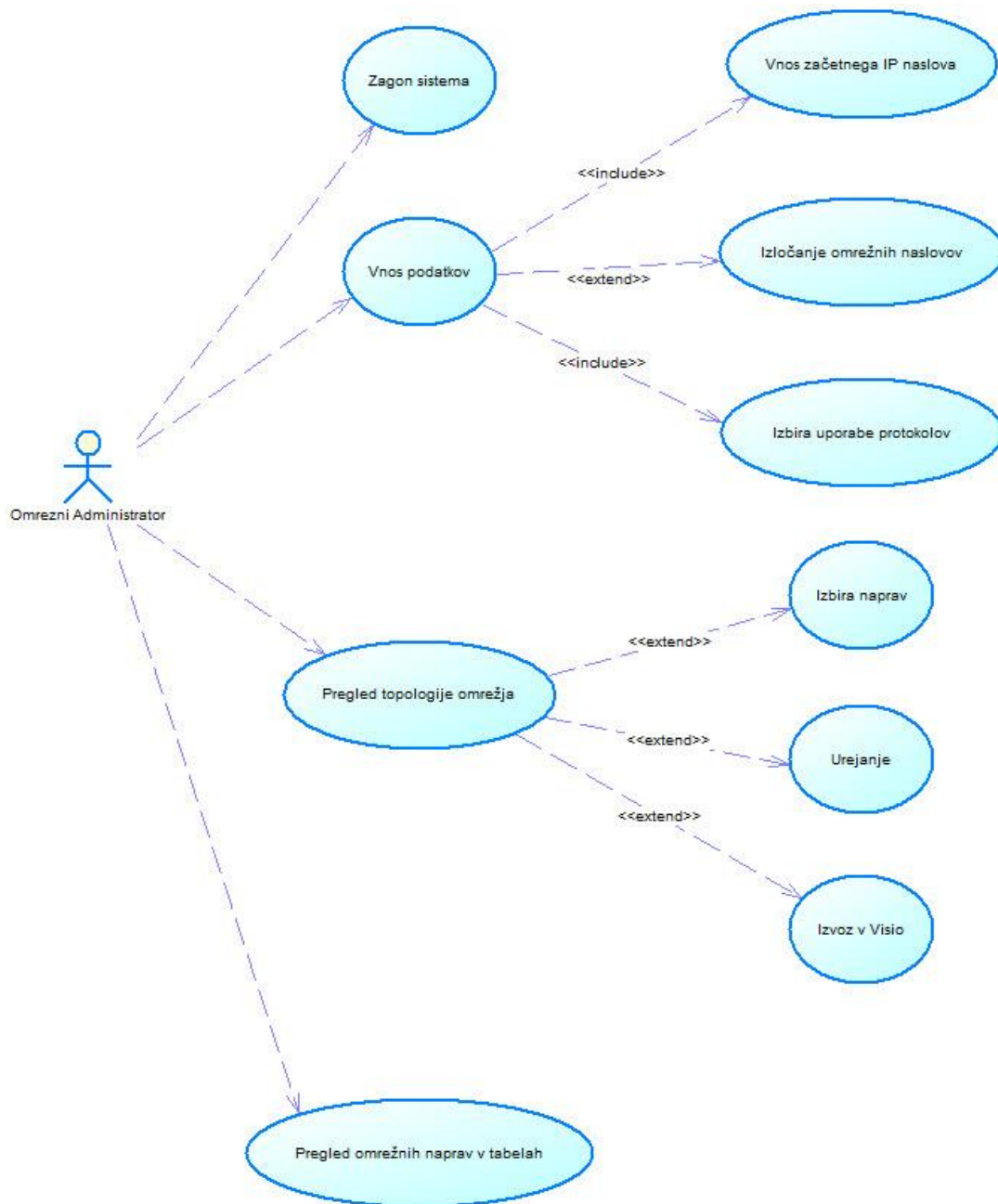
3.2.4 Tretja normalna oblika

Def: Relacija je v tretji normalni obliki, če je v prvi normalni in drugi normalni obliki in noben atribut, ki ni primarni ključ, ni tranzitivno odvisen od primarnega ključa [14].

Ker noben atribut v podatkovnem modelu, ki ni ključ relacije, tranzitivno ne določa drugega atributa, ki tudi ni ključ relacije, se naš podatkovni model nahaja v tretji normalni obliki.

3.3 Predstavitev programa

Razvoj programskega paketa je zahteval veliko sistematičnosti in predvidevanja težav, ki se lahko pojavijo pri izrisovanju topologije v velikih podjetjih, kjer imajo nekaj sto omrežnih naprav. Najpreprostejša predstavitev programskega paketa z vidika uporabnika (omrežnega administratorja) je diagram primerov uporabe [15]. Ta nam prikazuje vlogo uporabnikov in funkcije sistema. Diagram primerov uporabe je prikazan na Slika 13.



Slika 13: Prikaz diagrama primerov uporabe.

S Slika 13 je razvidno, da programski paket za izris omrežne topologije omogoča štiri osnovna dejanja:

- zagon sistema,
- vnos podatkov,
- pregled topologije omrežja,
- pregled omrežnih naprav v tabelah.

3.3.1 Zagon sistema

Po končanem razvoju programskega paketa za izris omrežne topologije smo ugotovili, da programski paket ne bo mogel delovati brez podatkovne baze. Ker podatkovna baza vsebuje procedure za vračanje, dodajanje in brisanje vrednosti v tabelah, smo se odločili, da bomo programski paket namestili na virtualni računalnik. Prednost tega postopka je v tem, da lahko uporabniku naložimo Microsoft SQL Server 2008 R2 in sprotno popravimo vse poti, da lahko programski paket deluje brez težav. Poleg teh prednosti je s pomočjo sistema virtualizacije uporabniku omogočen prenos programskega paketa za izris omrežne topologije v katerokoli okolje. Predvidevamo, da bi programski paket deloval v vseh okoljih, kjer je omrežje implementirano s pomočjo protokola OSPF in CDP ter je na vseh napravah omogočen protokol SNMPv1.

3.3.2 Vnos podatkov

S Slika 13 je razvidno, da je vnos podatkov razdeljen na tri dele:

- vnos začetnega IP naslova,
- izločanje omrežnih naslovov,
- izbira uporabe protokola.

Vnos začetnega IP naslova in izbira uporabe protokola sta obvezna. Pri prvem mora uporabnik programskega paketa napisati začetni naslov IP, od koder bo programski paket začel rekurzivno iskati naprave v omrežju. Na tem mestu je priporočljivo vpisati IP naslov naprave, ki je vključena v usmerjevalni protokol OSPF, saj programski paket deluje na način, da najprej rekurzivno pregleda vse sosednje naprave po protokolu OSPF, nato pa od vseh najdenih naprav pregleda še naprave po protokolu CDP. Na tak način onemogočimo, da bi program prehitro zaključil s pregledovanjem omrežja, saj se moramo zavedati, da protokol CDP deluje na drugi plasti in ni nujno, da vidi naprave, ki so med seboj povezane preko usmerjevalnih protokolov (npr. OSPF, RIP v2 in EIGRP).

Izločanje omrežnih naslovov je poljubna operacija vsakega uporabnika programskega paketa, zato je s Slika 13 razvidno, da je povezava do te operacije označena kot <<extended>>. Izločanje naslovov IP smo implementirali s predvidevanjem, da v večjih podjetjih ne želijo imeti določenih delov omrežja evidentiranih na sliki. To se navezuje tako na varnost kot tudi na morebitno lastništvo omrežja. Uporabnik v tabelo vnese območje IP naslovov, ki jih ne želi pregledovati, le ti pa se shranijo v podatkovno bazo, tako da uporabniku ni potrebno vedno znova vnašati vseh območij, ki jih želi izločiti. Urejanje teh naslovov bo prikazano v predstavitvi programskega paketa.

3.3.3 Pregled topologije omrežja

S Slika 13 je razvidno, da je pregled topologije omrežja razdeljen na tri dele:

- izbira naprav,
- urejanje slike omrežne topologije,
- izvoz v Visio.

Po končanem pregledu omrežja lahko uporabnik izriše celotno topologijo omrežja ali pa izbere samo en del. Po odločitvi se na komponento Microsoft Visio izriše omrežna topologija, ki jo mora uporabnik urediti. Kadar je na sliki veliko število omrežnih naprav in redundantnih povezav, mora uporabnik sam urediti sliko. Pred ali po končanem urejanju lahko uporabnik shrani sliko v Microsoft Visio in tam nadaljuje svoje delo z urejanjem omrežne topologije.

3.3.4 Pregled omrežnih naprav v tabelah

V tabelah lahko za vsako napravo pogledamo, katere sosede ima in preko katerih vmesnikov so priključeni nanjo. Tovrstne informacije so koristne kadar pride do težav v omrežju.

3.4 *Predstavitev implementacije programskega paketa za izris omrežne topologije*

Implementacija programskega paketa je nastajala skozi petmesečno obdobje, v katerem smo implementirali:

- uporabniški vmesnik,
- funkcijo za pregled omrežja,
- funkcijo za izris omrežne topologije.

3.4.1 Implementacija uporabniškega vmesnika

Implementacija uporabniškega vmesnika je nastajala postopoma. Na začetku razvoja programskega paketa smo sprva določili osnovno vizualno podobo, nato pa smo se preusmerili v implementacijo funkcij. Grafična predstavitev programa je opisana v poglavju 4.4.

3.4.2 Implementacija funkcije za pregled omrežja

Pri implementaciji funkcije za pregled omrežja smo implementirali funkciji za pregled naprav s simulacijo protokola OSPF in pregled naprav s simulacijo protokola CDP. Razlika med funkcijama je predvsem v hierarhičnih številkah za dostop do objektov, ki predstavljajo informacije o napravah. Za generiranje paketov SNMP smo uporabili knjižnico

SNMPSHARPNET [16] in funkcijo SNMPwalk. Slednja deluje na enak način kot funkcija z enakim imenom, ki je implementirana na operacijskem sistemu Linux.

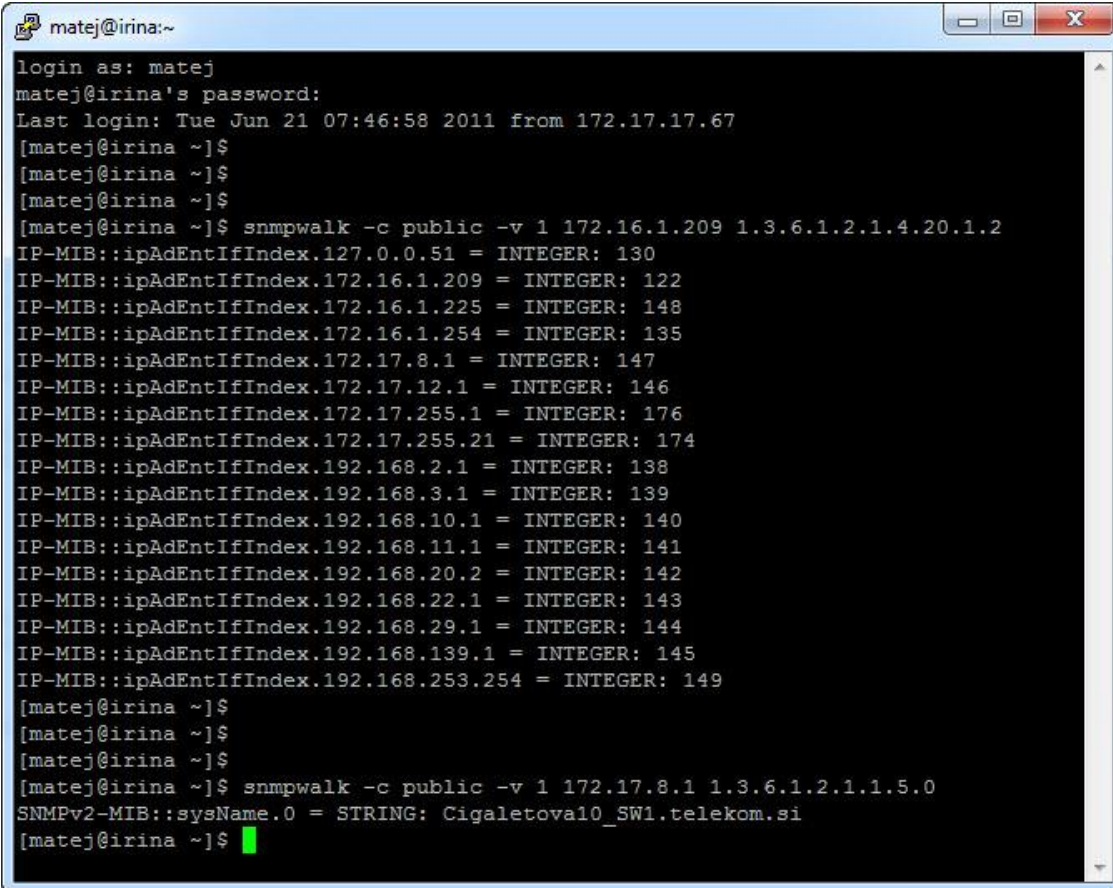
Postopek pregleda omrežja po protokolu OSPF je obsegal naslednje korake:

- pridobitev sosednjih naprav začetnega naslova IP,
- za vsak sosednji naslov IP smo s pomočjo vrednosti MIB: 1.3.6.1.2.1.1.5.0 (sysName) pridobili ime,
- po pridobitvi imena smo želeli pridobiti tudi izvorni in ponorni fizični vmesnik naprav,
- na koncu smo vsako napravo vnesli kot parameter v proceduro, ki je preverila ali naprava v podatkovni bazi že obstaja,
- postopek smo rekurzivno ponavljali na vseh napravah, dokler ni prišlo do ponovitve začetne kombinacije.

Ker je s pomočjo razvojnega okolja zamudno ugotavljati, kateri koraki so potrebni za pridobitev zelenih rezultatov, smo si pri tem delu pomagali s strežnikom, prikazanim na Slika 14. Ta je integriran v okolju, kjer smo razvijali programski paket za izris omrežne topologije. Slednji nam je omogočal uporabo funkcije SNMPWalk, ki smo jo uporabili za definiranje korakov pri ugotovitvi zaporedja poizvedb za pridobitev fizičnih vmesnikov na napravah po simulaciji protokola OSPF. Naslednji primer prikazuje zaporedje korakov, ki so potrebni za pridobitev imen fizičnih vmesnikov na ponorni (sosednji) napravi.

- s pomočjo vrednosti MIB: 1.3.6.1.2.1.4.20.1.2 (ipAdEntIfIndex) smo pridobili zaporedno številko vmesnikov,
- s pomočjo vrednosti MIB: 1.3.6.1.2.1.31.1.1.1.1 (ifName) smo pridobili ime vmesnika glede na zaporedno številko, ki smo jo pridobili v prvem koraku.

Strežnik, integriran v omrežje, nam je pomagal še pri mnogih drugih nalogah, saj je bilo mogoče ob vpisu naslova IP in vrednosti MIB pridobiti podatke z naprave.



```

matej@irina:~$ login as: matej
matej@irina's password:
Last login: Tue Jun 21 07:46:58 2011 from 172.17.17.67
[matej@irina ~]$
[matej@irina ~]$
[matej@irina ~]$
[matej@irina ~]$ snmpwalk -c public -v 1 172.16.1.209 1.3.6.1.2.1.4.20.1.2
IP-MIB::ipAdEntIfIndex.127.0.0.51 = INTEGER: 130
IP-MIB::ipAdEntIfIndex.172.16.1.209 = INTEGER: 122
IP-MIB::ipAdEntIfIndex.172.16.1.225 = INTEGER: 148
IP-MIB::ipAdEntIfIndex.172.16.1.254 = INTEGER: 135
IP-MIB::ipAdEntIfIndex.172.17.8.1 = INTEGER: 147
IP-MIB::ipAdEntIfIndex.172.17.12.1 = INTEGER: 146
IP-MIB::ipAdEntIfIndex.172.17.255.1 = INTEGER: 176
IP-MIB::ipAdEntIfIndex.172.17.255.21 = INTEGER: 174
IP-MIB::ipAdEntIfIndex.192.168.2.1 = INTEGER: 138
IP-MIB::ipAdEntIfIndex.192.168.3.1 = INTEGER: 139
IP-MIB::ipAdEntIfIndex.192.168.10.1 = INTEGER: 140
IP-MIB::ipAdEntIfIndex.192.168.11.1 = INTEGER: 141
IP-MIB::ipAdEntIfIndex.192.168.20.2 = INTEGER: 142
IP-MIB::ipAdEntIfIndex.192.168.22.1 = INTEGER: 143
IP-MIB::ipAdEntIfIndex.192.168.29.1 = INTEGER: 144
IP-MIB::ipAdEntIfIndex.192.168.139.1 = INTEGER: 145
IP-MIB::ipAdEntIfIndex.192.168.253.254 = INTEGER: 149
[matej@irina ~]$
[matej@irina ~]$
[matej@irina ~]$
[matej@irina ~]$ snmpwalk -c public -v 1 172.17.8.1 1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: Cigaletoval0_SW1.telekom.si
[matej@irina ~]$

```

Slika 14: Prikaz snmpwalk funkcije na strežniku, integriranem v podjetje.

Paket za poizvedbo smo ustvarili s pomočjo ukazov, ki so prikazani na Slika 15. Na začetku smo določili, da gre za protokol SNMP, verzijo 1. S pomočjo slednjega smo brali podatke iz naprave. V naslednjem koraku smo določili naslov IP naprave, od katere smo želeli pridobiti podatke. Na koncu je bilo potrebno vpisati vrednost MIB in v omrežje poslati zahtevo.

```

// SNMP community name
OctetString community = new OctetString("public");
AgentParameters param = new AgentParameters(community);
param.Version = SnmpVersion.Ver1;
IpAddress agent = new IpAddress(MasterDevice.DEVICE_IP);
UdpTarget target = new UdpTarget((IpAddress)agent, 161, 2000, 1);
Pdu pdu = new Pdu(PduType.Get);
pdu.VbList.Add("1.3.6.1.2.1.1.5.0"); //sysName
try
{
    // Make SNMP request
    SnmpV1Packet result = (SnmpV1Packet)target.Request(pdu, param);

    MasterDevice.DEVICE_NAME = result.Pdu.VbList[0].Value.ToString();
}
catch (Exception e)
{
}

```

Slika 15: Prikaz zaporedja ukazov za poizvedbo o imenu naprave.

Kot smo omenili v poglavju 3.1.3 je imela procedura, prikazana na Slika 16, glavno vlogo pri dodajanju omrežnih naprav v podatkovno bazo. Slednja skrbi za to, da ne prihaja do podvajanja kombinacije naprav v podatkovni bazi, saj bi to pomenilo glavno težavo pri izrisu omrežnih naprav na komponento MS Visio. Procedura preprečuje vnos kombinacije, kjer bi se zamenjala vloga glavne in končne naprave v omrežju.

```

ALTER PROCEDURE [dbo].[Insert_Network_Neighbour_Device]

@Master_Device_IP nvarchar(15),
@Neighbour_Device_IP nvarchar(15),
@Master_Device_Name varchar(100),
@Device_Name nvarchar(100),
@Device_Local_Int nvarchar(100),
@Device_Type nvarchar(100),
@Device_Platform nvarchar(100),
@Device_Destination_Int nvarchar(100),
@Scan_ID int

AS
BEGIN
IF NOT EXISTS(select * from Network_Device_IP_Range_Rules where dbo.fnBinaryIPv4(Start_IP_Range) <= dbo.fnBinaryIPv4(@Neighbour_Device_IP) and dbo.:
begin
IF NOT EXISTS(select * from Network_Device_Neighbour where Master_Device_IP=@Master_Device_IP and Neighbour_Device_IP=@Neighbour_Device_IP)
begin
IF NOT EXISTS(select * from Network_Device_Neighbour where Master_Device_IP=@Neighbour_Device_IP and Neighbour_Device_IP=@Master_Device_IP)
begin
insert into Network_Device_Neighbour
IF NOT EXISTS(select * from Network_Device_Neighbour where Master_Device_Name=@Master_Device_Name and Device_Name=@Device_Name)
begin
IF NOT EXISTS(select * from Network_Device_Neighbour where Master_Device_Name=@Device_Name and Device_Name=@Master_Device_Name)
begin
IF NOT EXISTS(select * from Network_Device where Device_IP=@Neighbour_Device_IP)
insert into Network_Device (Device_IP,Scan_ID) values (@Neighbour_Device_IP,@Scan_ID);

insert into Network_Device_Neighbour (Neighbour_Device_IP,Master_Device_IP,Master_Device_Name,Device_Name,Device_Platform,Device:
/*Zaradi števca naprav vrnemo vrednost, da smo vnesli v bazo novo napravo*/
Select 'Ok';
end
end
end
end
end
END

```

Slika 16: Prikaz procedure za vnos omrežnih naprav v podatkovno bazo.

3.4.3 Implementacija funkcije za izris omrežne topologije na komponento MS Visio

Za izris omrežne topologije na zaslon bi lahko uporabili odprtokodno komponento, vendar smo se zaradi velikega števila objektov na sliki odločili za komponento MS Visio [17]. Glavni prednosti slednje komponente sta stabilnost delovanja in možnost izvoza v program MS Visio. Stabilnost delovanja pripomore k zadovoljstvu končnih uporabnikov programskega paketa, medtem ko izvoz slike v program MS Visio omogoča končnemu uporabniku stalen pregled omrežne topologije na vseh računalnikih, saj lahko sliko pozneje pretvorimo v obliko pdf. Slabost te komponente je, da mora imeti uporabnik za delovanje našega programskega paketa nameščen program MS Visio.

Postavitev omrežnih naprav na komponento se izvede s pomočjo ukazov, prikazanih na Slika 17.

```
Microsoft.Office.Interop.Visio.Document currentStencil = axDrawingControl1.Document.Application.Documents.OpenEx(@"C:
(short)Microsoft.Office.Interop.Visio.VisOpenSaveArgs.visOpenDocked);
Microsoft.Office.Interop.Visio.Shape shape = currentPage.Drop(currentStencil.Masters["CISCO1401- custom 2"], x, y);
```

Slika 17: Prikaz ukazov, potrebnih za postavitev omrežne naprave na komponento MS Visio.

Za izris sosednjih naprav na krožnici glavne naprave smo naredili izračun za koordinati x in y:

- koordinata x = $X_{\text{(Glavne naprave)}} + \text{radij} * \text{Cos}(\text{kot})$,
- koordinata y = $Y_{\text{(Glavne naprave)}} + \text{radij} * \text{Sin}(\text{kot})$,
- kot smo izračunali s pomočjo formule $360/(\text{število sosednjih naprav})$.

3.5 Predstavitev programskega paketa za izris omrežne topologije

Dokumentacija opreme v velikih podjetjih predstavlja nočno moro za ljudi, ki s to opremo razpolagajo. Število omrežnih naprav v velikih komunikacijskih podjetjih presega število 200 ali več, poleg tega pa imajo naprave na kritičnih mestih redundantne povezave do ostalih naprav v omrežju. Ročni popis strojne opreme bi v velikem komunikacijskem podjetju trajal kar nekaj časa, poleg tega pa bi ravno redundantne povezave predstavljale težavo, saj je verjetnost zmotne pri ročnem popisovanju opreme precejšna. Poleg redundantnih povezav moramo biti pazljivi tudi na oblake MPLS, kjer je glavna naprava preko istega vmesnika povezana še z ostalimi napravami v omrežju. Do težav prihaja tudi zaradi že omenjene predstavitve posameznih usmerjevalnikov. Vzrok se skriva v dejstvu, da se lahko usmerjevalnik svojim sosedom predstavlja z različnim naslovom ID (angl. RouterID). Slednje prikazuje

Slika 18: Na sliki vidimo, da ima naprava: »172.17.8.1« za svojo sosednjo napravo: »172.16.1.210«. Ko poženemo isto poizvedbo na napravi »172.16.1.210«, vidimo da se naprava »172.17.8.1« predstavlja z naslovom »172.16.1.209«

, s katere je razvidno, da ima naprava z naslovom IP: »172.17.8.1« in imenom »Cigaletoval0_SW1« za eno izmed sosednjih naprav 172.16.1.210. Ko na slednji napravi izpišemo vse sosednje naprave, opazimo, da naslov 172.17.8.1 ni med izpisanimi. Med sosednjimi naslovi se namesto tega pojavi naslov 172.16.1.209, ki pripada napravi »Cigaletoval0_SW1«. Težavo lahko rešimo s tem, da poleg naslovov IP preverjamo še imena sosednjih naprav.

```

[matej@irina ~]$ snmpwalk -c public -v 1 172.17.8.1 1.3.6.1.2.1.14.10.1.1
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.210.0 = IPAddress: 172.16.1.210
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.211.0 = IPAddress: 172.16.1.211
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.212.0 = IPAddress: 172.16.1.212
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.213.0 = IPAddress: 172.16.1.213
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.214.0 = IPAddress: 172.16.1.214
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.215.0 = IPAddress: 172.16.1.215
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.216.0 = IPAddress: 172.16.1.216
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.226.0 = IPAddress: 172.16.1.226
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.227.0 = IPAddress: 172.16.1.227
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.228.0 = IPAddress: 172.16.1.228
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.229.0 = IPAddress: 172.16.1.229
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.230.0 = IPAddress: 172.16.1.230
SNMPv2-SMI::mib-2.14.10.1.1.172.17.255.22.0 = IPAddress: 172.17.255.22
SNMPv2-SMI::mib-2.14.10.1.1.192.168.2.5.0 = IPAddress: 192.168.2.5
SNMPv2-SMI::mib-2.14.10.1.1.192.168.2.6.0 = IPAddress: 192.168.2.6
SNMPv2-SMI::mib-2.14.10.1.1.192.168.2.7.0 = IPAddress: 192.168.2.7
SNMPv2-SMI::mib-2.14.10.1.1.192.168.2.8.0 = IPAddress: 192.168.2.8
SNMPv2-SMI::mib-2.14.10.1.1.192.168.29.50.0 = IPAddress: 192.168.29.50
SNMPv2-SMI::mib-2.14.10.1.1.192.168.253.251.0 = IPAddress: 192.168.253.251
SNMPv2-SMI::mib-2.14.10.1.1.192.168.253.253.0 = IPAddress: 192.168.253.253
[matej@irina ~]$ snmpwalk -c public -v 1 172.16.1.210 1.3.6.1.2.1.14.10.1.1
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.209.0 = IPAddress: 172.16.1.209
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.211.0 = IPAddress: 172.16.1.211
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.212.0 = IPAddress: 172.16.1.212
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.213.0 = IPAddress: 172.16.1.213
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.214.0 = IPAddress: 172.16.1.214
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.215.0 = IPAddress: 172.16.1.215
SNMPv2-SMI::mib-2.14.10.1.1.172.16.1.216.0 = IPAddress: 172.16.1.216
[matej@irina ~]$ snmpwalk -c public -v 1 172.16.1.209 1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: Cigaletova10_SW1.telekom.si

```

Slika 18: Na sliki vidimo, da ima naprava: »172.17.8.1« za svojo sosednjo napravo: »172.16.1.210«. Ko poženemo isto poizvedbo na napravi »172.16.1.210«, vidimo da se naprava »172.17.8.1« predstavlja z naslovom »172.16.1.209«

Pozitivna lastnost programa se pokaže tudi pri morebitni postopni menjavi naslovov IP ali preimenovanju naprav, saj program lahko vedno znova pregleda omrežje in na podlagi zbranih podatkov sestavi omrežno topologijo. Pri ročnem delu bi to pomenilo ponovni popis in spreminjanje naslovov na obstoječi sliki omrežne topologije.

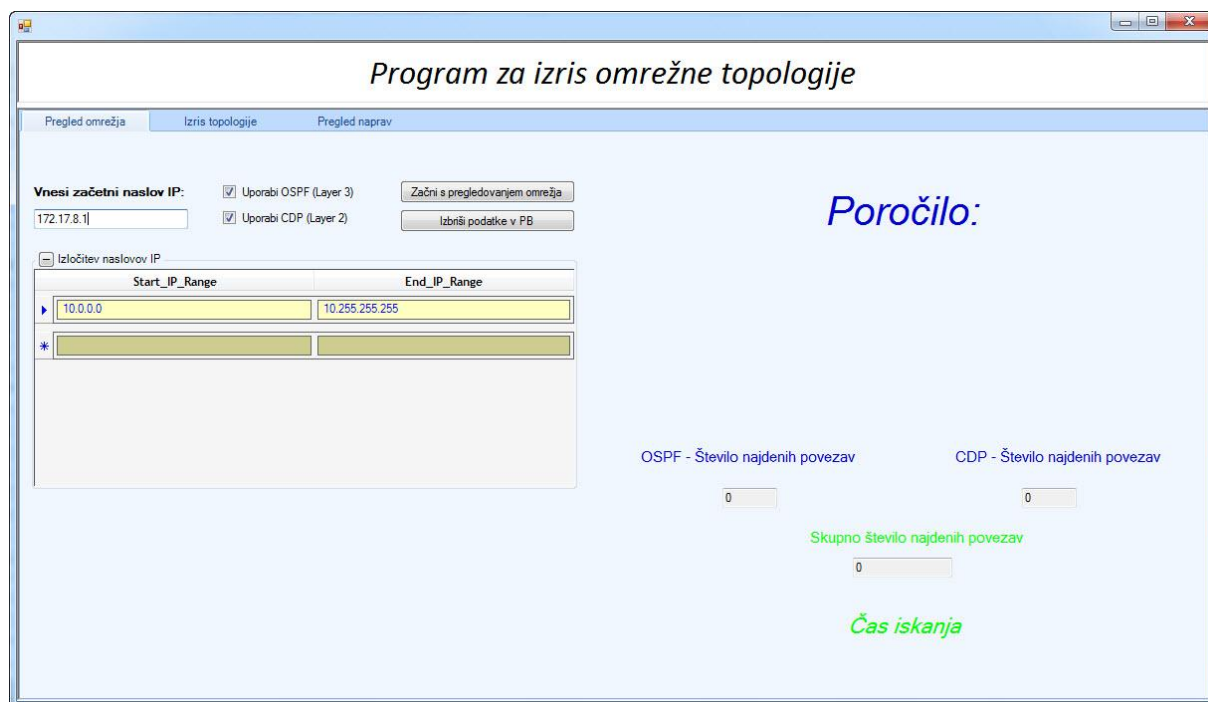
Programski paket za izris omrežne topologije je razvit s pomočjo razvojnega okolja Microsoft Visual Studio 2010, kjer smo uporabili programski jezik C#. Ta razvijalcem programske opreme omogoča, da se ne ukvarjajo z obliko uporabniških komponent, ampak te samo uporabljajo, v ozadju pa razvijajo logični del programskega paketa. Slednje nam prikazuje Slika 19, s katere je razvidno, da imamo na osnovni komponenti Panel nameščene tudi druge komponente (npr. TextBox, CheckBox, Label, Button, Grid). Vse te komponente razvijalcem omogočajo, da uporabljajo tako lastnosti kot tudi morebitne sprožilce ob kliku na izbrano komponento. Ker smo želeli uporabiti nekoliko boljše tabele in zavihke, smo uporabili nekaj komponent proizvajalca Infragistics [18].

3.5.1 Opis programskega paketa za izris omrežne topologije — pregledovalnik omrežja

Na Slika 19 je prikazan osnovni del uporabniškega vmesnika. Vidimo, da program vsebuje tri osnovne zavihke:

- zavihek za pregledovanje omrežja,
- zavihek za izris omrežne topologije,
- zavihek za pregled vmesnikov posameznih naprav.

V zavihku za pregledovanje omrežja se uporabnik sreča z vnosnim poljem, kjer mora določiti začetni naslov IP. Pomen tega polja je bil opisan v poglavju 3.3.2. Po končanem vnosu začetnega naslova IP mora uporabnik izbrati protokole, ki se bodo uporabljali pri pregledu, nato pa klikniti na gumb »Začni s pregledovanjem omrežja«, ki izvrši funkcijo za pregled omrežja. Med pregledom se aktivira polje »Poročilo«, kjer uporabnik pridobi informacije o številu najdenih povezav med napravami. To je posebej razdeljeno na najdene povezave preko simulacije protokola OSPF in najdene povezave preko simulacije protokola CDP.



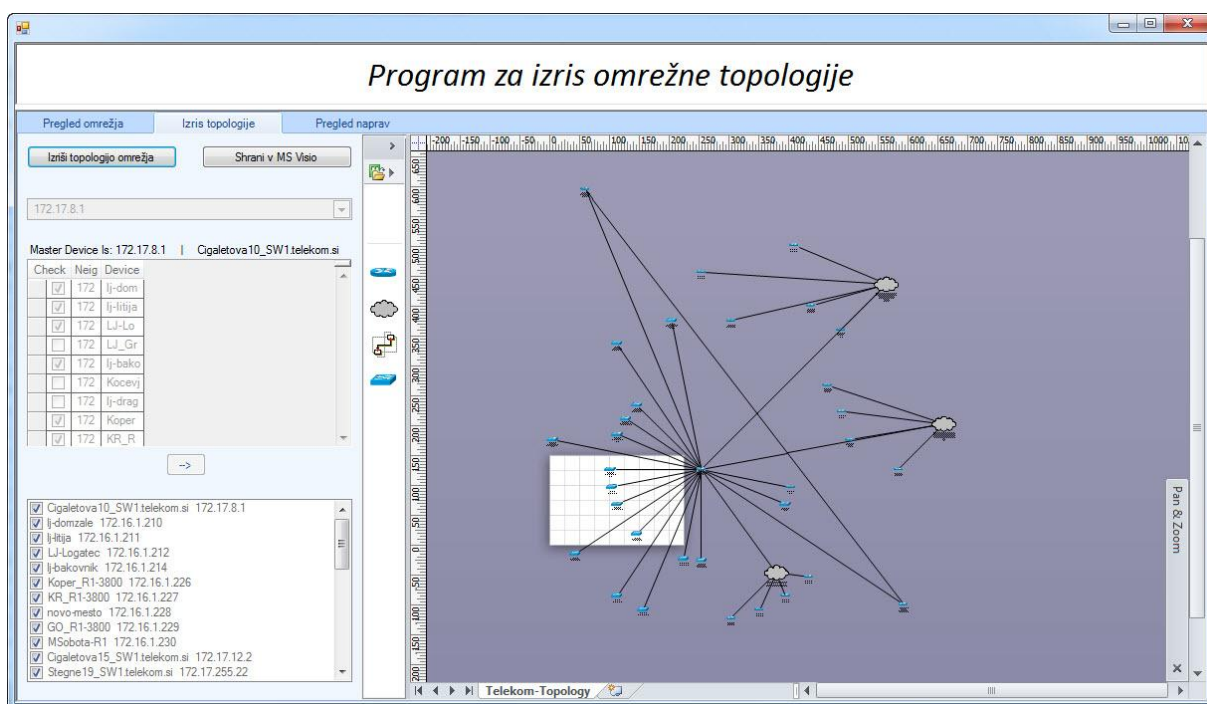
Slika 19: Prikaz programa in predstavitev zavihka za pregled omrežja.

Algoritem za pregled omrežnih naprav deluje tako, da programski paket najprej preveri vse sosednje naprave začetnega naslova IP, po protokolu OSPF. Ko programski paket pridobi vse sosednje naslove IP, za vsakega izvede poizvedbo, ki vrne ime omrežne naprave. Ko

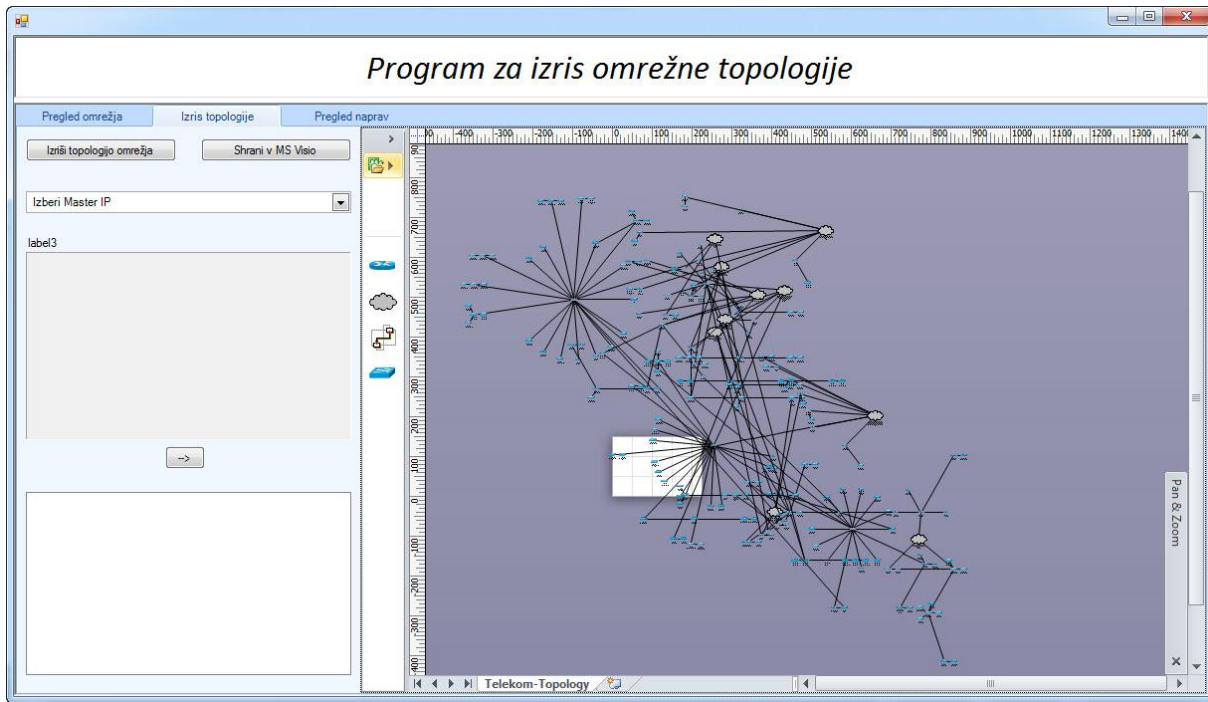
programski paket pridobi podatke o sosednjih napravah, jih vnese v podatkovno bazo, kjer procedura za vnos podatkov skrbi, da ne prihaja do podvajanja podatkov v podatkovni bazi. Po končanem pregledu omrežja nas programski paket obvesti o končanju, nato pa lahko uporabnik prične z izrisom omrežne topologije.

3.5.2 Izris omrežne topologije

Ta del programskega paketa je najpomembnejši, saj je tu dosežen namen razvoja le tega. S Slika 20 je razvidno, da programski paket za izris omrežne topologije vsebuje komponento, na katero izrisuje omrežne naprave in povezave med njimi [17]. Funkcije programskega paketa MS Visio so na tej komponenti zelo omejene. Namenjene so le pregledu, zato je priporočljivo, da po končanem izrisu izvozimo sliko omrežne topologije v programski paket MS Visio, nato pa tam urejamo sliko po lastni želji. Pri velikem številu naprav je odzivnost komponente MS Visio nekoliko slabša.



Slika 20: Prikaz zavihka za izris omrežne topologije.



Slika 21: Prikaz izrisa celotnega omrežja.

S Slika 21 je razvidno, da lahko uporabnik nariše celotno omrežno topologijo ali pa se odloči le za izris dela omrežja, kot je to prikazano na Slika 20. Za celotno omrežno topologijo uporabnik klikne na gumb »Izriši topologijo omrežja«, nato pa mora počakati nekaj časa, preden program prikaže rezultat na komponento MS Visio. Po končanem izrisu lahko uporabnik ureja ali izvozi omrežno topologijo v programski paket MS Visio. To naredi s pritiskom na gumb »Shrani v MS Visio«.

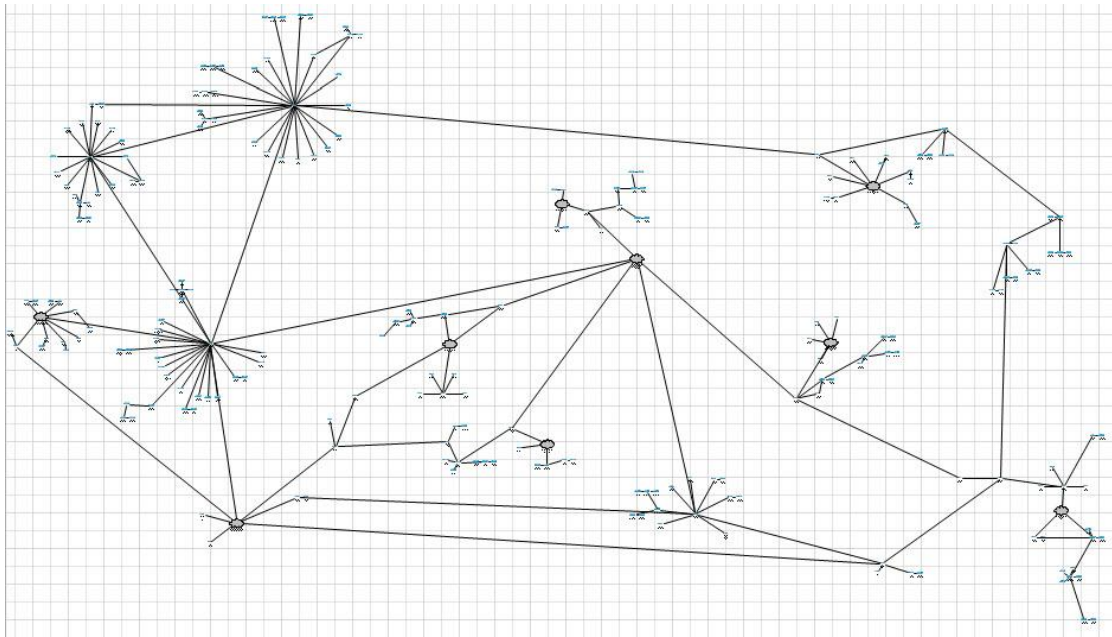
V drugem primeru lahko uporabnik izriše le usmerjevalnik s sosednjimi napravami. Slednje stori z izbiro usmerjevalnika s spustnega seznama in označitvijo sosednjih naprav, ki jih želi imeti prikazane na sliki. Izbor naprav prikazuje Slika 20. Po končanem izboru uporabnik klikne na gumb »Izriši topologijo omrežja« in počaka nekaj trenutkov, da program izriše sliko na komponento.

Algoritem za izris omrežne topologije deluje tako, da ustvarja skupine z napravami, ki sodijo skupaj. S Slika 10 je razvidno, da imamo dve vrsti naprav:

- »Network_Device_Master« (naprave s sosedi),
- »Network_Device_Neighbor« (končne naprave v omrežju).

Algoritem za izris poveže vse sosednje naprave z glavno napravo in ustvari skupino. Pri ustvarjanju skupine spusti vse sosednje naprave, ki so hkrati glavne naprave. Slednji pogoj nam razdeli celotno omrežno topologijo na več manjših skupin. Prednost tega algoritma je v tem, da je v primeru velikega števila naprav in povezav urejanje le teh enostavnejše. Urejena

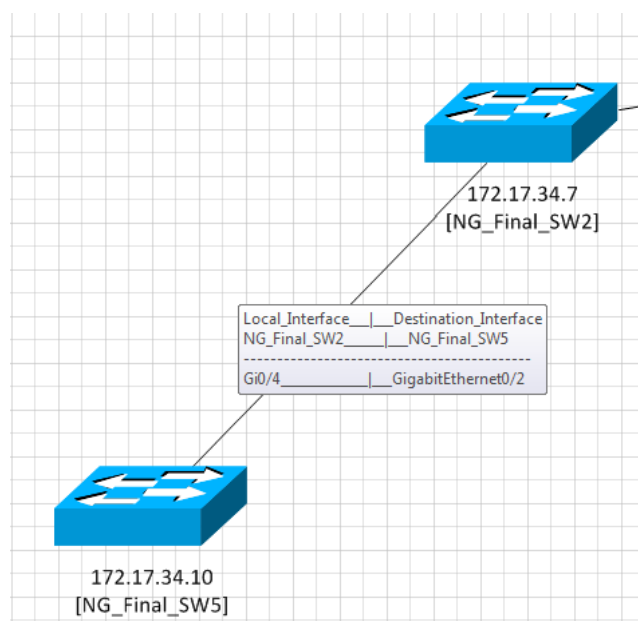
omrežna topologija večjega komunikacijskega podjetja je prikazana na Slika 22. S Slika 22 je razvidno, da podjetje povezuje svoje večje podružnice preko omrežja MPLS, poleg tega pa ima do glavnih central zagotovljene redundantne povezave.



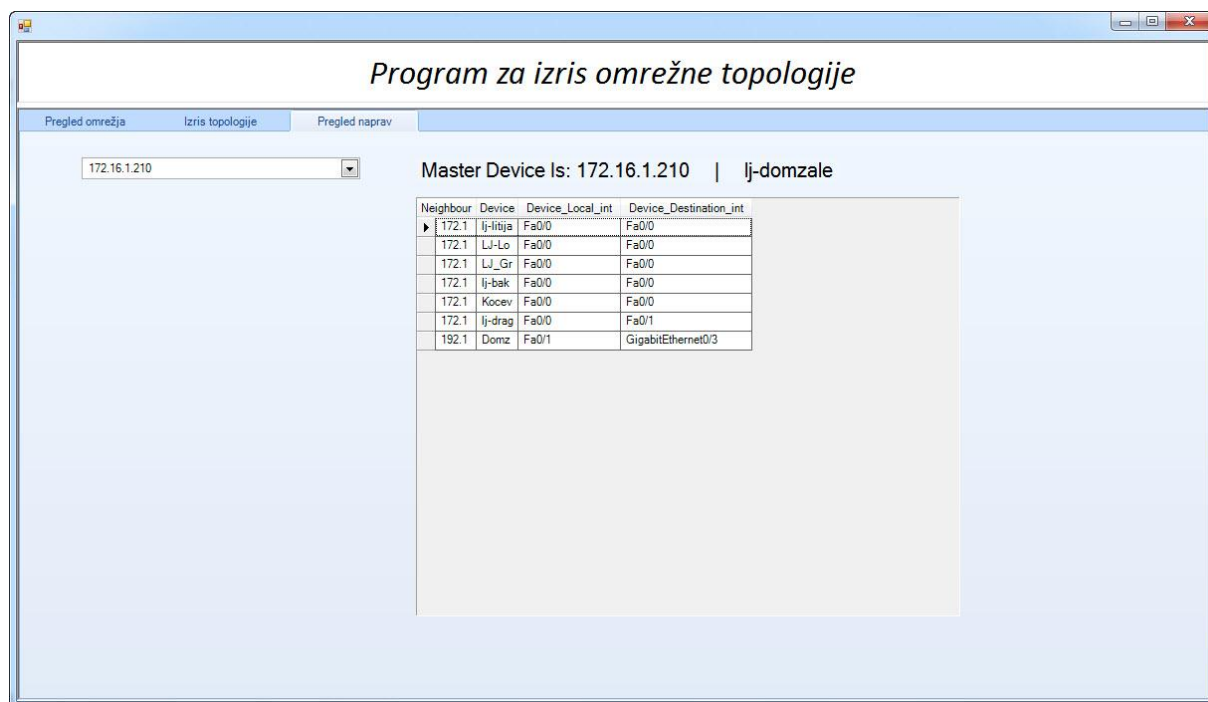
Slika 22: Prikaz urejene omrežne topologije.

3.5.3 Pregled vmesnikov posameznih naprav

Slednji zavihek je pomemben za omrežnega administratorja, kadar želi pridobiti informacije o povezavah med napravami. Te informacije lahko pridobimo tudi na sliki omrežne topologije, če je ta izvožena v programski paket MS Visio. Informacije so zapisane s pomočjo funkcije »Tooltip«, kar pomeni, da mora uporabnik na povezavi za nekaj trenutkov obdržati kazalec miške in funkcija bo poskrbela za prikaz okna z informacijami, kot je to prikazano na Slika 23. Okno vsebuje informacije o imenu in fizičnem vmesniku naprave.



Slika 23: Prikaz funkcije Tooltip.



Slika 24: Prikaz zavihka »Pregled naprav«.

Kadar uporabnik želi pregledati povezave na izbrani napravi, lahko to stori s pomočjo zavihka »Pregled naprav«, kot je to prikazano na Slika 24. Programski paket izpiše podatke o napravi v tabelo, ki je vsebovana v zavihku »Pregled naprav«. Tabela vsebuje podatke o:

- naslovu IP glavne omrežne naprave,
- naslovu IP končne omrežne naprave,

- imenu glavne naprave,
- imenu končne naprave,
- fizičnih povezavah na strani glavne naprave (»Local_Interface«),
- fizičnih povezavah na strani končne naprave (»Destination_Interface«).

4. Sklepne ugotovitve

Diplomsko delo zajema pregled in primerjavo obstoječih rešitev z na novo razvitim programskim paketom ter opis in predstavitev programskega paketa. Glavno težavo pri implementaciji programske rešitve je predstavljalo kompleksno okolje, ki je vsebovalo preko 3000 naprav z dodeljenim naslovom IP, od tega 200 omrežnih naprav, ki so bila med seboj povezana preko redundantnih povezav in omrežja MPLS ali ATM. Vse te naprave smo uspeli skozi obdobje razvoja povezati med seboj in izrisati s komponento Microsoft Visio. Algoritem za izris omrežne topologije uporablja kotne funkcije za izračun točk na krožnici in izris sosednjih naprav.

Za nadaljnji razvoj bi bilo potrebno spremeniti algoritem izrisovanja, tu glavno težavo predstavljajo povezave, ki se med seboj sekajo. Ko je teh sekajočih se povezav veliko, pride do nepreglednosti omrežne topologije. V zdajšnjem algoritmu za izris si mora uporabnik po izrisu vzeti nekaj časa in na svoj način urediti sliko. Poleg algoritma za izris bi bilo bolje, če bi programski paket pri iskanju naprav namesto imen upošteval fizične naslove vsakega vmesnika. Programski paket je pripravljen za prenos v različna okolja, vendar bi bilo kljub temu potrebno v drugih okoljih programski paket še popraviti in prilagoditi zahtevam ostalih okolij.

5. Kazalo slik

Slika 1: Prikaz razporeditve omrežnih naprav na tri nivoje [1].....	2
Slika 2: Prikaz odnosa agenta in upravljavca omrežja [2].	3
Slika 3: Hiearhično drevo ASN.1 za naslavljanje komponent v MIB [3].	4
Slika 4: Prikaz izpisa ukaza show cdp neighbors [8].	6
Slika 5: Prikaz izrisa omrežne topologije s pomočjo programskega paketa IPSwitch.....	7
Slika 6: Prikaz definiranja območja, ki naj ga program pregleda.	8
Slika 7: Prikaz različnih omrežij.	9
Slika 8: Prikaz načina povezave v večjem podjetju.	10
Slika 9: Prikaz konceptualnega modela podatkovne baze.....	13
Slika 10: Prikaz logičnega modela podatkovne baze.	15
Slika 11: Prikaz tabel v programskem paketu Microsoft SQL Server 2008 R2.....	16
Slika 12: Prikaz pretvorbe tabele "Network_Device_Neighbor" v drugo normalno obliko.	18
Slika 13: Prikaz diagrama primerov uporabe.	20
Slika 14: Prikaz snmpwalk funkcije na strežniku, integriranem v podjetje.	24
Slika 15: Prikaz zaporedja ukazov za poizvedbo o imenu naprave.....	24
Slika 16: Prikaz procedure za vnos omrežnih naprav v podatkovno bazo.	25
Slika 17: Prikaz ukazov, potrebnih za postavitev omrežne naprave na komponento MS Visio.	26
Slika 18: Na sliki vidimo, da ima naprava: »172.17.8.1« za svojo sosednjo napravo: »172.16.1.210«. Ko poženemo isto poizvedbo na napravi »172.16.1.210«, vidimo da se naprava »172.17.8.1« predstavlja z naslovom »172.16.1.209«.....	27
Slika 19: Prikaz programa in predstavitev zavihka za pregled omrežja.....	28
Slika 20: Prikaz zavihka za izris omrežne topologije.....	29
Slika 21: Prikaz izrisa celotnega omrežja.....	30
Slika 22: Prikaz urejene omrežne topologije.....	31
Slika 23: Prikaz funkcije Tooltip.	32
Slika 24: Prikaz zavihka »Pregled naprav«.....	32

Literatura:

- [1] (2011) Ulysses Ronquillo. Dostopno na:
<http://itdaddy.files.wordpress.com/2008/07/campusmodel.png?w=300&h=296>.
- [2] (2011) Podjetje Cisco. Dostopno na:
<http://www.cisco.com/en/US/i/000001-100000/35001-40000/35501-36000/35640.jpg>.
- [3] (2011) Wikipedia - MIB. Dostopno na:
<http://upload.wikimedia.org/wikipedia/commons/1/1c/SNMP.MIB-Tree.PNG>.
- [4] (2011) Podjetje Cisco. Dostopno na:
<http://tools.cisco.com/Support/SNMP/do/BrowseOID.do?local=en&translate=Translate&objectInput=1>.
- [5] (2011) Organizacija IETF. Dostopno na:
<http://tools.ietf.org/html/rfc5935>.
- [6] (2011) Wikipedia - OSPF. Dostopno na:
http://en.wikipedia.org/wiki/Open_Shortest_Path_First.
- [7] (2011) WikiPedia - CDP. Dostopno na:
http://en.wikipedia.org/wiki/Cisco_Discovery_Protocol.
- [8] (2011) n-study. Dostopno na:
<http://www.n-study.com/network/image/cdp3-thumb.gif>.
- [9] (2011) Wikipedia - MPLS. Dostopno na:
http://en.wikipedia.org/wiki/Multiprotocol_Label_Switching.
- [10] (2011) Gimnazija Maribor. Dostopno na:
http://www.tretja.si/informatika/4_letnik/teorija/Podatkovni_model_1.pdf.
- [11] (2011) Colos, Fakulteta za računalništvo in informatiko. Dostopno na:
http://colos1.fri.uni-lj.si/ERI/RACUNALNISTVO/NORMALIZACIJA/funkcionalne_odvisnosti.html.
- [12] (2011) Colos, Fakulteta za računalništvo in informatiko. Dostopno na:
<http://colos1.fri.uni-lj.si/ERI/RACUNALNISTVO/NORMALIZACIJA/1nf.html>.

[13] (2011) Colos, Fakulteta za računalništvo in informatiko. Dostopno na:
<http://colos1.fri.uni-lj.si/ERI/RACUNALNISTVO/NORMALIZACIJA/2nf.html>.

[14] (2011) Colos, Fakulteta za računalništvo in informatiko. Dostopno na:
<http://colos1.fri.uni-lj.si/ERI/RACUNALNISTVO/NORMALIZACIJA/3nf.html>.

[15] (2011) Univerza v Ljubljani. Dostopno na:
http://www.google.si/url?sa=t&source=web&cd=4&ved=0CCkQFjAD&url=http%3A%2F%2Ffmiha.ef.uni-lj.si%2F_dokumenti3plus2%2F192126%2FPO07.ppt&rct=j&q=diagram%20primerov%20uporabe&ei=dTUcTpeWGcPzsgawjpnhBg&usg=AFQjCNHjlsLilTsCa2shxuZ2C1G0AjfPXQ.

[16] (2011) SNMPsharpNET. Dostopno na:
<http://www.snmpsharpnet.com/>.

[17] (2011) Mike Gold. Dostopno na:
<http://www.c-sharpcorner.com/UploadFile/mgold/VisioInDotNet12032006222024PM/VisioInDotNet.aspx>.

[18] (2011) Podjetje Infragistics. Dostopno na:
<http://www.infragistics.com/>.