Нов Български Университет

департамент “Телекомуникации”

инж. Николай Петров Милованов

ТРАНСФОРМАЦИЯ НА МРЕЖИ И УСЛУГИ ОТ IPV4 КЪМ IPV6 В МРЕЖИТЕ ОТ СЛЕДВАЩО ПОКОЛЕНИЕ

ДИСЕРТАЦИЯ

за получаване на образователна и научна степен

“ДОКТОР НА НБУ“

професионално направление: 5.3 **“Комуникационна и компютърна техника”**

Нов Български Университет

департамент “Телекомуникации”

инж. Николай Петров Милованов

ТРАНСФОРМАЦИЯ НА МРЕЖИ И УСЛУГИ ОТ IPV4 КЪМ IPV6 В МРЕЖИТЕ ОТ СЛЕДВАЩО ПОКОЛЕНИЕ

ДИСЕРТАЦИЯ

за получаване на образователна и научна степен

“ДОКТОР НА НБУ“

професинално направление: 5.3 **“Комуникационна и компютърна техника”**

Научен ръководител:

доц. д-р Иван Богомилов

Благодарности

Изказвам сърдечни благодарности на научния ми ръководител доц. д-р. Иван Богомилов, че ме въведе в тази интересна и актуална научна област, за дискусиите при обсъждането на резултатите, представени в настоящата дисертация, а също така и за полезните съвети при изготвянето и оформлението й.

Изказвам благодарности на:

Съпругата ми Весела и на дъщеричката ни Андреа за безрезервната им подкрепа. Специални благодарности на Андреа за това, че не избърза и се появи с написването на последните редове на дисертационния труд.

проф. Антони Славински за плодотворното ни сътрудничество и за създаването на добра работна атмосфера в департамент „Телекомуникации“ и като цяло в „Нов Български Университет“

проф. Маргарита Петкова за първоначалното ми въведение в докторантурата, за съвместните ни дискусии на тема „Какво се изисква от мен като Докторант“ и за изготвянето на първоначалния ми работен план.

проф. д-р David Garlan за възможността да се включа в академичния живот на Institute for Software Research, Carnegie Mellon University, за участието в научна работна група ABLE “Architecture Based Languages and Environments” и в семинара SSSG “Software Research Seminar” и за многобройните ни разговори на тема “IP network evolution”.

д-р. Георги Петров, Биляна Бенкова и колегите от Департамент Телекомуникации за многобройните ни дискусии, за административната подкрепа и за съвместна работа по проекта I3E.

д-р. Васил Йорданов за многобройните ни дискусии и безусловната подкрепа при разработката на софтуерния прототип.

д-р. Георги Шарков и екипа на ESICenter за предоставената ми възможност да отида на специализация в Carnegie Mellon University.

Специални благодарности на членовете на работна група “4to6trans”, с които дискутирахме проблемите при прехода от IPv4 към IPv6 и дефинирахме нуждата от софтуерни средства за подпомагане процеса.

Съдържание

Увод, цели и структура на дисертационния труд

Еволюцията на телекомуникационните мрежи е процес, продиктуван от непрекъснато променящите се нужди на съвременното информационно общество. Процесът винаги е свързан с назряването на определени нужди, появата на нови технологии и приложението на технологиите в мрежите на различни по големина доставчици на съдържание и услуги. Процесът на преход от една технология към друга може да се случи стихийно, според дадена стратегия или в комбинация от двете. В съвременните телекомуникационни мрежи този процес не е непознат, често се случва да се мигрира от една технология към друга по-модерна, по-бърза и в крайна сметка по-добра. Примери за подобни миграции са подмяната на един вид релейни технологии с други, подмяната на медни кабели с оптични, замяната на множество преносни технологии от слой две на OSI (Open System Interconnection) [1] модела с Ethernet. Общото между всички тези промени е, че са локални и винаги се отнасят до подмяната на една или друга технология под или над нивото на Internet Protocol (IP) протокола.

IP е възникнал като протокол, който да обедини създадените до този момент мрежи за данни. Целта е била да бъдат свързани мейнфрейм машините на водещи научни, военни и бизнес организации в САЩ (Съединените Американски Щати), за да може обединения им изчислителен ресурс да се използва за разработката на сложни научни експерименти. Постепенно мрежата, използвайки силните страни на IP протокола, е еволюирала до две отделни мрежи – Milnet и Internet. Milnet и до момента се използва за военни цели, а Internet е мрежата за глобална комуникация. През годините Internet и мрежите за данни, базирани върху IP са заели важна социално икономическа роля в съвременното общество. Повечето от нас едва ли биха могли да си представят живота без информационни услуги като email, google, www, facebook и skype. Бизнес корпорациите едва ли биха могли да функционират без мрежата, свързваща отделните им офиси или без мрежата, даваща достъп на милиони клиенти до услугите, предоставени от тях.

Силата на Интернет е пряко свързана с качествата на неговото най-силно звено – Интернет протокола. Основното технологично предимство на IP е възможността да работи с всяка една преносна технология под него като в същото време може да пренася всякакъв вид полезна информация. Информацията, която може да бъде пренесена в един IP пакет е

практически неограничена и зависи само и единствено от максималния размер на сегмента на преносната среда.

Основният недостатък на протокола е свързан с един от неговите най-важни компоненти – адресното пространство. За разлика от информацията, адресите, използвани в Интернет имат ограничен размер – 32 бита или четири байта. В началото на 70-те години това е било считано за огромно число, много по голямо от бройката на супер-компютрите по онова време. Още с бума на модемните технологии през 80-те и най-вече през 90-те години става ясно, че това адресно пространство няма да е достатъчно за нуждите на бързо разрастващата се Интернет екосистема. В резултат на това се появяват няколко предложения за промяна на текущата версия четири на Интернет протокол. Надделява версия шест, специфицирана през 1995 в RFC 1883 [2].

Новата версия на Интернет протокол предлага много по-голямо адресно пространство и редица други подобрения. Единственият проблем е, че новата версия не е съвместима със старата. Постепенно се появяват редица механизми за преход от IPv4 към IPv6, а също така и механизми, които да позволят многократното използване на IPv4 адресно пространство.

Въпреки наличието на множество механизми за преход 18 години след създаването на IPv6 масовия преход към новия протокол все още не е факт [3]. Редица водещи телекоми споменават наличието на желание и наличието на стратегия за преход, но такъв и до момента не е изпълнен [4].

Според [4], [5] има множество причини за това. Преходът би бил скъп и не носи директни от бизнес гледна точка за оператора, който го извършва. Преходът е труден и изисква промени в технологията, на която се основава съществуващата мрежа. Изпълнението му без специализиран софтуер, който да подпомогне техническия персонал, изглежда като невъзможна задача в контекста на голям доставчик на мрежови услуги [6]. Преходът е рискован и има реална опасност съществуващите услуги да бъдат засегнати и операторът да не успее да изпълни част от договорните клаузи с определени клиенти [7], [8].

Анализирайки подобни източници може да се заключи, че е трудно да бъдат оценени бизнес и техническите критерии, които са предпоставка за избор на една или друга

стратегия за преход и че дори и да бъде избрана дадена стратегия е изключително трудно тя да бъде приложена.

В настоящата дисертация авторът предлага подход за решаване на проблема. Подходът е естенсвена еволюция на методите за преход разгледани в [7] [9] и е естествено надграждане над предишна разработка на автора представена в [10]. Авторът разглежда прехода от IPv4 към IPv6 като процес на еволюция на комуникационната мрежа от едно текущо състояние в друго желано. Еволюционният преход зависи от контекста на дадената мрежа и от разбиранията, и желанията на различните заинтересовани от дадената мрежа лица.

Преходът може да се изпълни чрез изпълнение на редица стъпки. Всяка една стъпка се състои от:

- технически ограничения, които трябва да бъдат налице, за да бъде възможно изпълнението;
- бизнес ограничения, които служат за оценка на стъпката;
- действие, което променя текущото състояние на мрежата;
- ефект, който служи за проверка дали резултата от изпълнението на стъпката е този, който е очакван.

Стъпките могат да бъдат групирани в стратегии, а стратегиите могат да бъдат оценени на базата на бизнес и техническите ограничения на съставните им стъпки.

Трансформацията на мрежата от текущото към желаното състояние се извършва по стратегията, която отговаря изцяло на техническите ограничения и съвпада най-точно с бизнес интересите на различните заинтересовани лица.

Съвременните комуникационни мрежи са достатъчно сложни и поради това вече нищо не може да се прави на ръка [6]. Голяма част от проблема по-прехода към IPv6 реално се свежда до липсата на средства за автоматизирано и контролирано изпълнението на стъпките от стратегиите. Авторът предлага решение и на този проблем като създава и прототип на софтуерна система способен да:

- да разкрие и генерира на топологичен модел от данни, съдържащ най-важните архитектурни мрежови свойства на текущото състояние на дадена „жива“ мрежа;
- да показва топологията на различни нива от OSI модела на даденото състояние;
- да генерира и визуализира еволюционните разлики между произволни две състояния и да подпомогне мрежовите архитекти в процеса на проследяване на еволюцията на техните мрежи в дълъг период от време;
- да изпълни контролирано и автоматизирано стъпките от стратегиите за преход между две състояния.

Глава 1: Обзор на мрежовите технологии, свързани с прехода от IPv4 към IPv6

В първа глава е направен литературен обзор на различни технологии, свързани с IP мрежите и с прехода към IPv6. Главата започва с хронология на Интернет и продължава с обзор и анализ на IPv4, IPv6 и механизмите за автоматизирано раздаване на адреси.

Обзорът на мрежовите технологии завършва с преглед на механизмите за преход от IPv4 към IPv6. Механизмите са условно разделени в три групи – механизми, работещи на база превод на адреси, механизми, работещи чрез изграждане на тунели и използване на двоен IP стек. Анализирани са предимствата и недостатъците на всеки един от механизмите за преход.

1.1 Интернет - история и откриватели

Началото на Интернет е поставено в САЩ (Съединените Американски Щати) през втората световна война от учен на име Vannevar Bush. Той създава няколко агенции, обединяващи учени с представители на бизнеса и държавата. Целта е била реализацията на важни военни, научни проекти. Самите агенции и дейността им са били засекретени, но това не попречило на Bush да стане знаменитост. Директна връзка между Bush и Интернет няма, но има няколко индиректни. От една страна Bush, освен че очевидно е бил далновиден ръководител, е и учен с доста модерни за времето си идеи. Една от тях, така наречения “memex”, е машина, която може да съхранява големи обеми от информация, а потребителите на “memex” са можели да извличат информация от нея и да я използват за различни цели. Години след това е създаден hypertext (хипертекста), Google и се стигнало до идеята информацията да се съхранява в дадена мрежа, така че да е възможен отдалечен достъп от много точки до нея.

Другата връзка между Bush и Интернет е ARPA (Advanced Research Projects Agency). ARPA е създадена през 1957 г. в САЩ в отговор на изстрелването на Спутник - първия съветски спътник. Основната цел на агенцията е Съединените Щати да бъдат водеща сила в областта на новите технологии. Скоро след създаването на агенцията се изгражда мрежата ARPAnet, считана за основоположник на съвременния Интернет. [11]

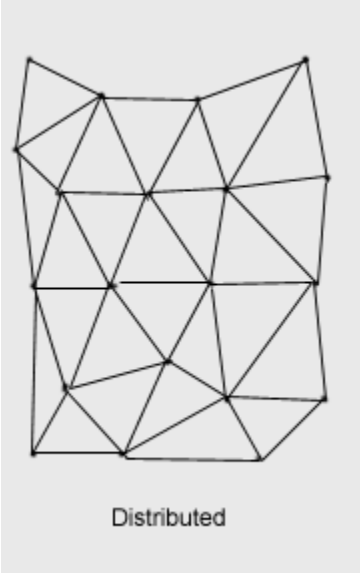
Идеята за “memex” и дейността на ARPA са много важни фактори за създаването на Интернет, но далеч не са единствените. Тук е мястото да бъдат споменати имената на J.C.R. Licklider, Bob Taylor и Paul Baran, изиграли важна роля в тази посока [11].

През 1963 Licklider е ученият, който първи изказва и защитава тезата, че компютърните системи могат да се използват за симулация на човешкото мислене. Реално той предлага създаването на мрежа между компютрите на институтите, работещи с ARPA.

Bob Taylor, директор на отдела на IPTO (Information Processing Techniques Office) в периода 1966-1969, е един от хората, възприели идеите на Licklider за мрежа между компютри и реално полага административното начало на проекта. Той осигурява финансирането, необходимо за осъществяването на ARPAnet и назначава Lary Roberts за главен архитект и ръководител на екипа, по създаването на мрежата. Архитектурата, използвана от Roberts, е базирана на разработка на Paul Baran.

Paul Baran се счита за бащата на пакетната комутация. Разработките му са базирани на задание, поставено от американската армия, за създаване на мрежа с високо ниво на надеждност и сигурност, която да може да издържи ядрена атака. За целта Baran предлага използването на пакетна комутация, голям брой мрежови възли и множество връзки между тях, т.е това, което днес наричаме разпределена мрежа [12].

Фигура 1-1 Разпределена мрежова архитектура, предложена от Баран (Източник [12])



В оригиналния си вариант ARPAnet е мрежа, която свързва няколко американски университета с ARPA. Първоначално Lary Roberts и екипът му стартират ARPAnet с четири възела (nodes) – University of California, Stanford Research Institute, University of Utah и UC Santa Barbara.

В техническо отношения мрежата е предоставила свързаност между големите меинфрейм машини чрез използването на по-малки компютри, наречени IMPs (Interface Message Processors) - Фигура 1-2. IMP-та изпълнявали ролята на комуникационни шлюзове или с други думи са първите routers, или както са наречени на български - маршрутизатори.

Фигура 1-2 Първият маршрутизатор [11]

Interface Message Processor (IMP)

Не след дълго започнали да се появяват и други мрежи подобно на ARPA - SATNET, ALOHANET, а в самата ARPA били включени множество нови възли. Появява се и идеята за обединение на различните мрежи в една обща. Тогава възниква и проблема с унифициране на комуникационните протоколи. Всяка мрежа използвала свой собствен протокол и реално обединението било трудно осъществимо [13].

За да се получи единна мрежа от множеството мрежи, била нужна още една стъпка - разработването на нови протоколи и технологии, които да бъдат стандартизирани и достъпни за всички. Една от тези технологии е Ethernet, създадена от Robert Metcalfe, а

друга е протоколния стек TCP/IP (Transmission Control Protocol/Internet Protocol), създаден от Vint Cerf и Bob Kahn. Протоколният стек TCP/IP и Ethernet са основните инструменти на съвременния Интернет и до днес. С право Vint Cerf е наречен „Баща на Интернет“ [13]. Идеята информационния поток да бъде разделен на сегменти, които да се енкапсулират в пакети, пренасяни в рамки (Данни->TCP/UDP->IP->Ethernet) се оказала толкова добра, че се използва в почти непроменена форма и до днес.

С появата на новата мрежа се разкрили нови възможности по отношение на достъпа до съдържание. Пионерите тук са:

- Ted Nelson - описва за първи път понятието хипертекст. Това е система от свързани документи, която по-късно ще се трансформира в WWW (World Wide Web) [14].
- Tim Berners-Lee - физик от CERN (Conseil Européen pour la Recherche Nucléaire) фактически поставя началото на WWW. WWW прави споделянето и достъпа до информация по-лесно и интуитивно от всеки един досегашен протокол [14]. Скоро след това Интернет се утроява, удесеторява и започва да се развива в една доста по-глобална насока от първоначалните цели на ARPAnet.
- Marc Andreesen - Създава първия WWW браузър с графичен интерфейс.

В резултат от усилията на тези и много други пионери от началото на 80-те години от миналия век, Интернет вече бил най-утвърдената мрежа и почти всеки университет или научен институт притежавал компютър, свързан с него.

1.2 Хронология

Освен цитираните по-горе имена има и още много други хора, допринесли за създаването и развитието на Интернет. Самото развитие е дълбоко свързано с развитието с еволюцията на човешкото общество и на комуникацията между хората като цяло.

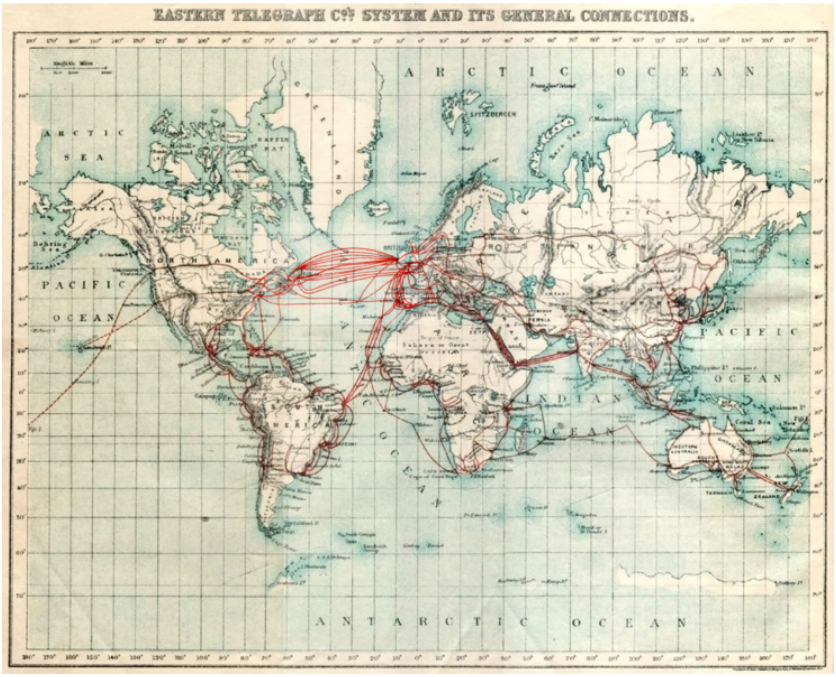
За начало се приема периода, в който древните народи са открили нуждата от пренос на информация на разстояние. За целта са ползвали птици, димни сигнали и други инструменти.

Francesko Lapi - В далечната 1536 година използва за пръв път символа “@” в писмо [11].

През 1830, Joseph Henry демонстрира предаване на електрически сигнал по метална жица с дължина 1 миля.

През 1835, Samuel Morse подобрява изобретението на Henry и създава морзовия код и налага телеграфа като световна система за комуникация.

Фигура 1-3 Викторианският Интернет (1901) [15]



През 1875г. Alexander Bell чува звук на другия край на телеграфната жица и поставя началото на нова ера в телекомуникациите – открит е телефона и е реализирано предаване на звукова информация по кабелна преносна система [12].

В периода 1895-1896г. е поставено началото на радиокомуникациите и са конструирани първите радиопредаватели и радиоприемници от Маркони и Александър Попов.

Joseph John Thomson открива електрона през 1897г. В последствие това откритие се оказва в основата на цяла една технологична революция, довела в крайна сметка и до създаването на Интернет.

През 1937г. John Atanasov и Clifford Berry започват работа по първия електронен цифров апарат. През 1942г. първият компютър е вече факт.

1945г. - Vannevar Bush предлага разработването на мемекс и поставя началото на първите агенции, обединяващи бизнес, наука и правителство [13].

През 1951г. се появява Ferranti Mark 1, първият „комерсиален“ компютър. До 1957г. са произведени още осем такива.

1956г. - IBM създава първия твърд диск. Директна връзка между твърдия диск и Интернет няма, но без него Интернет в сегашната му форма едва ли би могъл да съществува.

1957г. - СССР изстрелва Спутник. Скоро е създадена ARPA и е поставено началото на началото на Интернет.

1961г. - Leonard Kleinrock публикува първата научна статия, засягаща пакетната комутация.

1963г. - Licklider развива идеята, че компютрите трябва да са свързани в мрежа.

1964г. - Paul Baran доразвива идеята за пакетна комутация и предлага създаването на разпределени мрежи.

1965г. - Ted Nelson за пръв път използва думата hypertext. Скоро след това е изпратено електронно съобщение, което може да се счита за първия email. Gordon Moore прогнозира, че компютрите ще удвояват мощността си на всеки 18 месеца. Тази прогноза е известна като „Закон на Moore“ и е валидна и до днес. Това не е учудващо като се има предвид, че през 1968г. Moore създава Intel - световния лидер в разработката и производството на процесори.

1966г. - За първи път е предадена информация по оптично влакно. През същата година Bob Taylor получава финансиране за създаването на ARPAnet.

1968г. - Lary Roberts пуска първия RFQ (Request for Quotation) за IMP-та. RFQ документът е отхвърлен от IBM и HP с аргумента, че подобно устройство не може да бъде създадено. В крайна сметка търгът е спечелен от BNN - малка консултантска компания. След около година са създадени първите рутери.

1969г. - Steve Crocker създава първия RFC (Request For Comment) документ (RFC 1). През същата година BNN инсталира първите четири IMP-та и поставя реалното начало на ARPAnet.

1971г. - ARPAnet нараства. Възлите са вече 15. RFC документите вече са достигнали числото 172.

1972г. - Създадена е програма, позволяваща изпращането на email съобщения по ARPAnet.

1973г. - Robert Metcalfe създава Ethernet. Скоро след това той и David Boggs ще реализират първия Ethernet контролер и мрежа от точка до точка с космическата за тогава скорост от 2.944 Mbps. Години по-късно Robert Metcalfe създава 3COM. През същата година Ken Thompson и Dennis Ritchie показват на света Unix - операционната система, повлияла много на Интернет и информационните технологии от тогава и до сега.

1974г. - Vint Cerf и Bob Kahn публикуват "A Protocol for Packet Network Internetworking" – статия, описваща TCP. Това е също така и първият документ, в който се споменава думата Интернет. По късно Vint Cerf не случайно е наречен „баща на Интернета".

1977г. - Steve Jobs и Steve Wozniak създават Apple. Скоро след това те разработват и първия персонален компютър в гаража на Jobs. През същата година са произведени и продадени първите модеми. Междувременно ARPAnet вече наброява 111 крайни устройства.

1978г. – IP е отделен от TCP. През същата година е изпратен и първия спам (непоискано електронно съобщение). Става ясно, че email протоколът има недостатъци в тази посока.

1979г. - ARPA, която от скоро се казва DARPA (Defence Advanced Research Projects Agency), създава ICCB (Internet Configuration Control Board). Новата организация има за цел да подпомага процеса на свързване между мрежата и крайните устройства (тогава компютри).

1980г. - ARPAnet спира да функционира след като едно IMP получава дефект и „заразява" маршрутизиращите таблици на останалите IMP-та с невярна информация.

1981г. – Създадена е препоръка RFC 791 - Internetwork Protocol version 4.

1982г. - Протоколният стек TCP/IP е избран за използване в ARPAnet и всички останали военни мрежи на американската армия с военна директива. Проектът реално е осъществен през следващата година.

През същата година Scott E. Fahlman предлага комбинацията от символите „:-)“ като индикатор за усмивка и добро настроение в електронните съобщения.

1983г. - Компютрите в ARPAnet надвишават 500. Интернет става реалност след като ARPAnet е разделена на военна и цивилна. Цивилната подмрежа е това, което днес наричаме Интернет. През същата година Paul Mockapetris публикува RFC 882 и RFC 883, които описват DNS (Domain Name Service). DNS позволява асоциация между адрес и име в Интернет. Епохално откритие - ако не беше Paul Mockapetris, все още щяхме да пишем IP адресите в браузърите.

През 1983г. ICCB е преименувано в IAB (Internet Advisory Board). През същата година са създадени и множество групи, които целенасочено да развиват отделни аспекти на Интернет. С това е поставено началото на Обществото на Интернет Инженерите (IETF - Internet Engineering Task Force).

1987г. - Устройствата в Интернет вече са 10000.

1989г. - Интернет нараства до 100000 устройства. Дефиниран е първия Web проект. Tim Berners-Lee предлага протокол за споделяне на информация между научните тимове в CERN (Conseil Européen pour la Recherche Nucléaire).

1990г. - Tim Berners-Lee създава първия “прародител” на съвременния браузър.

1991г. - Броят на крайните възли е 600000. ARPAnet престава да съществува. Остават само Интернет и Milnet.

1992г. - В Интернет вече има повече от 1000000 крайни устройства. Създадени са първите текстови браузъри. IAB разбира, че IPv4 има фундаментален недостатък - адресното пространство е твърде недостатъчно за темповете, с които се разраства Интернет. Поставено е началото на създаването на IPv6.

1993г. - Участниците в Интернет достигат 2 милиона. Mark Andreessen създава първия графичен браузър. Създаден е WWW.

1995г. - HTTP (Hypertext Transfer Protocol) трафикът заема доминиращата позиция в Интернет. Броят крайни възли вече е 4 милиона. Представено е RFC 1883 - Internet Protocol Version 6.

1996г. - Устройствата в Интернет са вече 9 милиона. Lary Page и Sergey Flin създават търсачката Google.

1997г. - Устройствата в Интернет са 16 милиона. Създадена е ARIN (American Registry for Internet Numbers). Регистриран е милионния домейн запис (bonnyview.com).

1998г. – ITU (International Telecommunications Union) стандартизира протокола за пренос на данни чрез модем V.90. Той се използва в 56к модемите, направили достъпа до Интернет възможен за всеки, който има телефон. През същата година IEEE (Institute of Electrical and Electronics Engineers) публикува и стандарта за GigabitEthernet. През 1998г. е създадена и асоциацията ICANN (Internet Corporation for Assigned Names and Numbers), която чрез локалните си подразделения, отговаря за Интернет адресите и домейните в Интернет.

1999г. - Започва да се говори за IoT (Internet of Things) - Интернет мрежа не само от компютри и комуникационни устройства, но и от милиони (милиарди) сензори, датчици, интелигентни контролери и други микроелектронни устройства.

2000г. – Регистрирани са 304 милиона Интернет потребители и 10 милиона домейн имена. ICANN е принудена да пусне освен традиционните .com, .net, .org и домейн суфикси като .aero, .biz, .coop, .info, .museum, .name, .pro.

2011г. - Раздадени са последните IPv4 адресни блокове. Дори в България 40% от населението се преброява по Интернет. Умира Steve Jobs. Преди това създава милиони устройства със странни имена като iPhone и iPad, имащи възможност за комуникация с Интернет. Internet of Things започва да се превръща от идея в реалност. IPv6 се превръща в необходимост.

1.3 Интернет протокол

1.3.1 Интернет Протокол версия 4 (IPv4)

Интернет протокол версия 4 (IPv4) е в основата на съвременните мрежи. Това е протоколът, без който информационните потоци и услугите, които всеки е свикнал да използва не биха били същите. За пръв път е бил специфициран през далечната 1980 година в RFC 760 [16]. Скоро след това през 1981г. се е появила нова спецификация RFC 791 [17], която допълва спецификацията на протокола. От тогава до сега са направени множество добавки на IPv4 и на съпътстващите го протоколи от TCP/IP протоколния стек. Променяни са основно протоколите в слоя под IP или над IP. Другият тип промени е свързан с промяна на предназначението на някои от полетата на самия протокол. Примери за подобни са полетата Options или Type of Service.

IP е платформата, върху която е изграден Интернет. Протоколът се е доказал като най-гъвкавия и адаптивен механизъм за пренос на информация между две точки в съвременния свят. Разбира се, в началото никой не си е поставял такива цели. Протоколът е бил създаден, за да задоволи потребността на американската армия за пренос на информация. В момента, в който Интернет мрежата и информацията предлагана в нея е станала достатъчно популярна и потребителите на тази информация достатъчно многобройни е излязъл наяве най-съществения недостатък на Интернет протокол версия четири (IPv4) – ограниченото адресно пространство. Изследвания на множество независими източници показали, че при тогавашните темпове на развитие на световната икономика, то може напълно да се изчерпи до 2005г. На база на подобни изследвания, Интернет общността е взела две много важни и взаимно противоречащи си решения за гарантиране растежа на Интернет и Интернет обществото.

Решение номер едно е въвеждането на механизми за преобразуване на адресите. По този начин зад един или няколко публични IPv4 адреса могат да се крият неограничен брой потребители с частни IP адреси.

Решение номер две е да започне работа по нова версия на протокола. Новата версия първоначално е била обект на ожесточен спор между привържениците на два враждуващи лагера:

1. Привържениците на минималната промяна смятали, че протоколът не трябва да се променя, а само трябва да се разшири полето на адресното пространство. Основната теза на тази група била да се минимизират ресурсите необходими за разработка на устройства.
2. Привържениците на втората група смятали, че в IPv4 има доста неща за променяне и би било по целесъобразно да се направи нова версия на протокола. Тя трябвало да отговори на нарасналите нужди за адресно пространство, да избегне недостатъците на текущата версия четири и да послужи като платформа за развитие на бъдещия Интернет.

Интернет обществото предпочело разработката на нов протокол с името IP Next Genration (IPng). Скоро това име било заменено с IPv6, което си е останало и до днес. Архитектите на новият протокол са Steven Deering и Robert Hinden. Първата препоръка, специфицираща визията за новия протокол е издадена в края на 1995г. под номер RFC 1883 [2].

След като се е появил RFC 1883, започнал период на трескаво очакване за първите продукти и услуги, базирани на IPv6. Първите стъпки били трудни, новият протокол не носел все още никакви конкретни ползи, а в същото време Интернет обществото изживявало така наречения ".com" бум. Компаниите били изправени пред следната дилема: Докато инвестициите в IPv6 можели да донесат ползи в бъдеще, инвестициите в процъфтяващия през тези години IPv4 Интернет генерирали приходи на момента. Поради това не е изненада, че повечето от компаниите се съсредоточили върху продуктите базирани върху IPv4, обещаващи бързо и лесно възвръщане на инвестициите.

През този период основните радетели и изследователи на IPv6 били университети и правителства на определени държави. Лидери в това отношение са азиатски държави, като Япония, Корея и Китай, които имали сравнително малък дял от IPv4 адресното пространство на глава от населението в сравнение с държавите от Северна Америка и Европа. В Европа протоколът се развивал, основно на базата на проекти финансирани от Европейския съюз. Примери за такива са 6INIT [18], Euro6IX [19], 6NET [20] и 6DEPLOY [21].

Въпреки множеството проекти до 2000г. почти никой не използвал IPv6. След това „.com” балонът се спукал и Интернет обществото започнало да обръща все по-голямо внимание на новия протокол. В периода от 2000г. до 2004г. повечето операционни системи и производители на мрежово оборудване добавили поддръжка на IPv6 към продуктите си. Дали причината за това се корени в Европейските проекти, натиска на азиатските държави или на факта, че американската армия спряла да купува оборудване, неподдържащо IPv6 можем само да гадаем. Важното е, че в днешни дни почти няма операционни системи или мрежови устройства, които да не поддържат IPv6.

Остава въпросът „Защо тогава преходът все още не се случва и малцина са хората и организациите използващи реално IPv6?“.

1.3.2 Интернет Протокол версия 6 (IPv6)

Скоро след създаването си, първоначалното RFC 1883 [2], било подменен от RFC 2460 [22], а по-късно и от други RFC-та [23]. Прави впечатление, че привържениците на първата група са загубили битката и от самото начало новата версия на Интернет протокола е несъвместима със старата. Тя предлага много по-голямо адресно пространство, интегрира много от протоколите, свързани с IPv4 в себе си и предлага редица новости. Механизмите за препредаване на пакети се променят, OPTIONS полето изчезва, механизмите за раздаване на IP адреси и за откриване на съседи също претърпяват коренни промени. С други думи IPv6, както и IPv4 преди 20 години, предлага почти всичко необходимо за едно ново развитие на съвременните телекомуникационни и интернет мрежи. Това, което протоколът не предлага и не налага, е стратегия и средства за преход от версия четири към версия шест. В резултат на липсата на стратегия за това, как да бъде извършен прехода, вече повече от десет години след първоначалната поява на IPv6 все още по-малко от един процент от Интернет е преминал към новия протокол.

1.3.3 DNS (Domain Name System)

DNS е протокол за определяне на адреса, отговарящ на дадено име [24]. Той играе важна роля при IPv4, а също и при IPv6. Потребителите са свикнали да изписват имена вместо адреси. При IPv6 DNS [25] ще играе още по-важна роля тъй като адресът е по-дълъг и по-сложен за изписване от този при IPv4. DNS работи на база на описанието на всяко едно име в зона. Взаимовръзката на IP адресите и имената се описва като част от

дадената зона. Например, ако домейн името е xxx.com, то ще бъде описано в зона xxx.com. Зоната може да съдържа различни записи. Най-често използваните полета са CNAME, A,AAAA,MX, NS.

- CNAME (Canonical Name) - свързва Псевдоним (Alias) с асоцииранoто име (например www.xxx.com с xxx.com).
- A – свързва FQDN (Fully Qualified Domain Name) име с IPv4 адрес (xxx.com - X.X.X.X).
- MX – записът за маршрутизация на email съобщенията съдържа A/AAAA запис и приоритет. Използва се за насочване на email съобщенията към конкретни email сървъри, част от съответната зона (например MX 10:xxx.com, където 10 е приоритета, а xxx.com е A запис).
- AAAA - свързва FQDN с IPv6 адрес (xxx.com – X:X:X::X).
- NS – съдържа запис, оказващ местоположението на DNS сървъра/рите за съответния домейн.

DNS протоколът играе основна роля при прехода от IPv4 към IPv6. За да бъде успешна дадена миграционна стъпка не е достатъчно само да бъде мигрирана мрежовата инфраструктура, а също така трябва да бъдат конфигурирани отново съответните DNS зони. Като първа стъпка трябва да се добавят AAAA записи до съответните A записи.

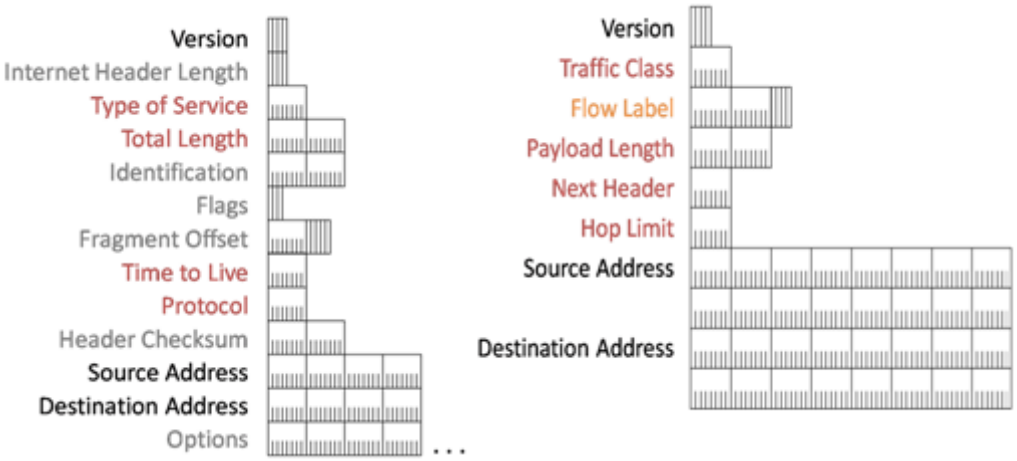
Друга допирна точка са част от механизмите за преход, които работят на базата на DNS заявки като добавят префикси пред A записа и го преобразуват в AAAA [26]. За целта е добавена специална функционалност към DNS известна като DNS64.

1.3.4 Съпоставка между IPv4 и IPv6

1.3.4.1 Заглавни части

Интернет протоколът е основния мрежов протокол на слой 2 от TCP/IP протоколния стек. На Фигура 1-4 са представени заглавните части на четвъртата и шестата версия на протокола [16], [2].

Фигура 1-4 IPv4 и IPv6 заглавни части



Основните полета са:

- **Version** – версия на протокола. При IPv4 има стойност 4, а при IPv6 6.
- **Type of Service (IPv4)/ Traffic Class (IPv6) –** полета, използвани за маркиране на трафика и прилагане на качество на обслужване. В зависимост от маркировката съществуват множество.
- **Flow Label (IPv6) –** използва се за идентификация на пакети, които принадлежат на един поток.
- **Total Length(IPv4)/ Payload Length (IPv6) –** съдържа число равно на размера на дейтаграмата при IPv4 или при IPv6 това е размера на полезната част на дейтаграмата в байтове, включително и допълнителните заглавни части.
- **Next Header (IPv6) –** това поле показва вида на следващите данни (следващ етикет или протокол от по-висок слой като TCP или UDP).
- **Time to Live (IPv4)/ Hop Limit (IPv6) –** и двете полета се използват за налагане на рестрикции относно „живота" на дейтаграмата в мрежата. Страната изпращач попълва дадена стойност, която намалява с едно на всеки възел по пътя към получателя. Ако стойността стане 0, дейтаграмата се отстранява и се изпраща ICMP съобщение на страната изпращач. Това поле предпазва Интернет мрежите от зацикляне на трафика. Ако се получи такова, поне е сигурно, че пакета ще може да мине през ограничен брой възли.
- **Source Address (IPv4) (IPv6) -** адрес на страната изпращач. Съдържа 32 битов адрес при IPv4 и 128 битов при IPv6.

- **Destination Address -** адрес на страната получател. Съдържа 32 битов адрес при IPv4 и 128 битов при IPv6.

Заглавната част на IPv6 дейтаграмата има олекотен формат в сравнение с IPv4. Въпреки че, IPv6 адресът е 4 пъти по-голям от този при IPv4, цялата заглавна част е по-малка и е фиксирана на 40 байта. Това се постига чрез премахване на множество ненужни полета. Полета, налични при IPv4 и отпаднали при IPv6 са:

- **Internet Header Length** – отпаднало поради фиксираната дължина на заглавната част;
- **Identification, Flags, Fragmentation Offset** – тези три полета се използват при фрагментация на IP дейтаграмите и при IPv6 са преместени в отделен, допълнителен етикет;
- **Header Checksum** – протоколите, пренасящи IP дейтаграмите от слой две имат подобно поле и проверяват за CRC грешки, следователно подобно поле е излишно в протокол на слой три;
- **Options** – полето Options бива заменено от допълнителните етикети.

1.3.4.2 Допълнителни етикети (Extension headers)

Допълнителните етикети са една от основните разлики между IPv4 и IPv6. Те се добавят към полезната част на IPv6 дейтаграмата, след основната заглавна част само, ако това е необходимо. Техният брой може да варира и поради тази причина е въведен механизъм за скачване на етикетите. Механизмът е реализиран чрез полето „Next Header“ (Следващ етикет). То има две основни задачи – да определи следващия етикет във веригата или ако няма такъв, да посочи следващия протокол (т.е. протоколът от по-горно ниво). В този случай функцията на полето е идентична на тази от полето „Протокол“ в IPv4. Полето се съдържа както в основната заглавна част, така и във всяка от допълнителните такива. Последният допълнителен етикет винаги определя протокола от следващото ниво на TCP/IP протоколния стек. Таблица: 1-1 съдържа стойностите на полето „Next Header“ за някои от най-често използваните допълнителни етикети, а Таблица: 1-2 за най-често използваните протоколи.

Таблица: 1-1 Стойности на „Next Header“ за най-често използваните допълнителни етикети

Стойност	Допълнителен етикет
0	Hop-by-hop option
43	Routing
44	Fragment
50	Encapsulating Security Payload (ESP)
51	Authentication Header (AH)
59	No next header
60	Destination Option
62	Mobility Header

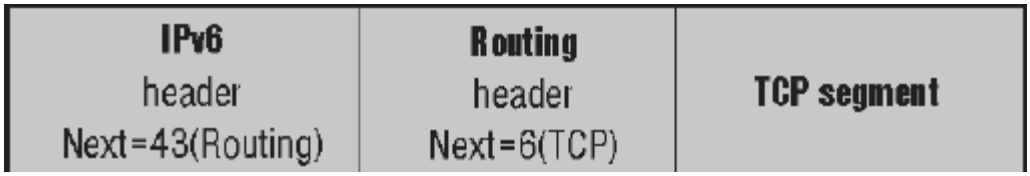
Таблица: 1-2 Стойности на „Next Header“ за най-често използваните протоколи

Стойност	Протокол
6	TCP – Transmission Control Protocol
8	EGP – Exterior Gateway Protocol
9	IGP – Interior Gateway Protocol
17	UDP – User Datagram Protocol
46	RSVP – Resource Reservation Protocol
47	GRE – Generic Routing Encapsulation
58	ICMP – Internet Control Message Protocol

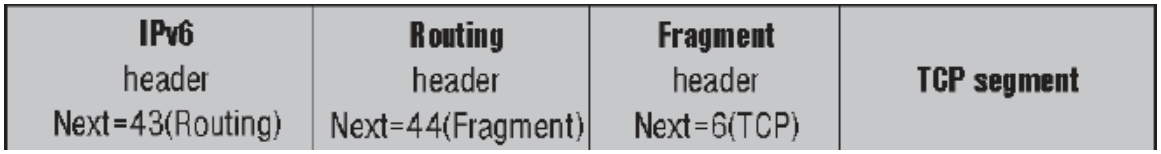
Механизмът с допълнителните етикети има предимства и недостатъци. От една страна той позволява олекотен формат на основната заглавна част на IPv6 пакета, а от друга може да доведе до дълга и сложна за декодиране поредица от допълнителни етикети. Това би затруднило обработката на дейтаграмите в транзитните възли. Поради тази причина е въведено групиране на допълнителните етикети. Етикетите, които са от значение за всички възли са подредени преди тези, които засягат само крайните получатели. По този начин междинните възли не е нужно да обработват всички допълнителни етикети, а това допринася за по-голямо бързодействие. Примери за IPv6 дейтаграми с/без допълнителни етикети са демонстрирани на Фигура 1-5, Фигура 1-6 и Фигура 1-7.

Фигура 1-5 IPv6 Дейтаграма без допълнителни етикети

Фигура 1-6 Дейтаграма с маршрутизиращ допълнителен етикет



Фигура 1-7 Пример за повече от един прикачени етикети



1.3.5 IPv6 адреси - основни принципи и видове

Бързото изчерпване на публичните IPv4 адреси е основната причина за разработването и въвеждането в експлоатация на IPv6. В настоящата подточка са разгледани принципите, видовете и механизмите за избор на адрес при IPv6 спрямо [23].

1.3.5.1 Основни принципи

След като адресното пространство при IPv4 (32 bits) явно се е оказало недостатъчно, при IPv6 съвсем логично то е значително разширено. IPv6 адресът независимо от вида си е 128 битов (16 bytes) или с други думи е черти пъти по-голям от този на предшественика си [23]. Тъй като всеки бит реално удвоява бройката на адресите, всъщност размерът на адресното пространство при IPv6 е огромен. То съдържа 2^{128} или 340 милиарда, милиарда, милиарда уникални адреси, което вероятно би стигнало за достатъчно дълъг период от развитието на човечеството.

За разлика от IPv4 при IPv6 е прието адресите да се представят с 32 шестнадесетични цифри [23]. За прегледност цифрите са подредени в 8 групи от по 4 числа. Групите са

отделени една от друга с двоеточия. По този начин един IPv6 адрес изглежда по следния начин:

Фигура 1-8 Класически запис на IPv6 адрес

```
2001:0718:1c01:0016:020d:56ff:fe77:52a3
```

За улеснение е прието водещите нули във всяка група да не се изписват. По този начин адреса горе всъщност може да се изпише и така:

Фигура 1-9 Съкратен запис на IPv6 адрес

```
2001:718:1c01:16:20d:56ff:fe77:52a3
```

Прието е също група, състояща се само от нули да се заменя с двойно двоеточие. Във всеки адрес може да има само по едно такова. Ако има повече от едно двойно двоеточие се счита, че имаме невалиден адрес. Това е особено полезно при изписването на специални адреси (loopback) и адреси, съдържащи дълги поредици от нули. Например следния loopback адрес 0:0:0:0:0:0:0:1 може да се изпише и така ::1.

Прието е всеки мрежови адрес да има префикс и маска prefix::/length. Префиксът съдържа стойностите на важните битове от адреса, а дължината показва броя на важните битове от началото на адреса. Тъй като останалите битове не играят определена роля, те се заменят с нули и съответно се представят с едно двойно двоеточие. Например префиксът отделен за транслация от IPv6къмIPv4 е 2002::/16. Това означава, че първите 16 бита от адреса трябва да съдържат шестнадесетичното число 2002, а останалите битове са незначителни.

Не всички адреси се третират по един и същи начин. IPv6 поддържа три различни типа адреси, от които зависи как ще бъде доставен пакета [23].

- **Unicast (индивидуален) address –** идентифицира един единствен уникален мрежови интерфейс (например компютър). Пакетите се доставят само на него.
- **Multicast (групов) address –** идентифицира група от интерфейси. Пакетите се доставят на всички в групата.
- **Anycast (селективен) address –** идентифицира група от мрежови интерфейси. Пакетите се доставят само на най-близкия член на групата.

При IPv6 липсват характерните за IPv4 Broadcast адреси. Те са заменени със специални групови адреси. Например ff02::1 е груповия адрес, идентифициращ всички интерфейси, свързани към даден мрежови сегмент.

В следващите подточки са разгледани в детайли отделните типове IPv6 адреси.

1.3.5.2 Unicast (Индивидуални) адреси

Индивидуалните адреси са така наречените нормални адреси, идентифициращи всеки компютър, сървър или интерфейс в една мрежа. Според RFC 3587 има няколко вида unicast IPv6 адреси [27].

Първата основна група са така наречените глобални (global) индивидуални адреси. Те се състоят от три части.

Фигура 1-10 Unicast IPv6 адрес

Global routing prefix е така наречения при IPv4 мрежови префикс. Той идентифицира уникално дадена мрежа в Интернет адресното пространство.

Subnet ID – идентифицира подмрежите в дадена мрежа. Всяка мрежа може да бъде разделена на множество подмрежи.

Interface ID – съдържа идентификатор на мрежовия интерфейс, част от дадена мрежа/ подмрежа. Идентификаторите на интерфейси са уникални за всяка подмрежа. Съвсем възможно е едно устройство да има един и същи идентификатор за различни подмрежи. Интернет стандартите изискват така наречения EUI-64 (Extended Unique Identifier) да играе ролята на идентификатор на интерфейса.

Реално всеки интерфейс има 48 битов глобален префикс, 16 битова подмрежа и 64 битов идентификатор на интерфейса. Поради това най-често глобалните IPv6 адреси изглеждат по следния начин.

Фигура 1-11 Unicast IPv6 адрес с 16 битова подмрежа и 48 битов Global routing prefix

Не всички IPv6 адреси са глобални. Някой от тях са ограничени само до един единствен физически (layer 2) сегмент. Тези адреси се наричат link-local (локални адреси за дадена линия) и се обозначават с префикса fe80::/10. Те се използват единствено за комуникация между интерфейсите в дадения физически сегмент. Маршрутизаторите не трябва да препредават пакети с link-local адрес на получателя. Тези адреси се използват за осъществяване на някои нови механизми въведени в IPv6 като например автоматична конфигурация на мрежовите параметри на даден интерфейс. В RFC 3513 са дефинирани два вида локани адреси: link-local с префикс fe80::/10 и site-local с префикс fec0::/10. Тъй като дълго време не се е постигнал консенсус какво точно трябва да бъде "site", RFC 3879 отменя "site" префиксите и запазва fec0::/10 за бъдеща употреба.

1.3.5.3 Идентификатор на интерфейса – EUI-64

EUI-64 е 64 битово поле от IPv6 адреса, което играе ролята на уникален идентификатор на интерфейса. 64 бита за идентификация на един интерфейс в един layer 2 (L2) сегмент изглежда твърде разточително като се има предвид, че 48 бита са напълно достатъчни за същата цел при Ethernet. При това в Ethernet, адресът е уникален не само за сегмента, но за всички сегменти. Една от причините за това разточителство е, че този идентификатор играе важна роля при автоматичната конфигурация на мрежовите параметри на интерфейса.

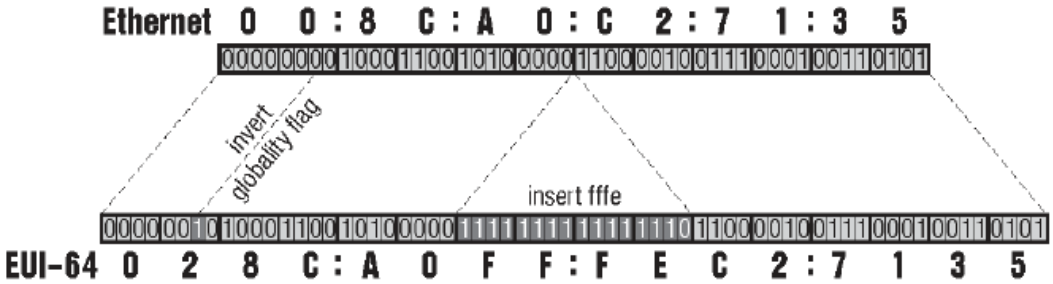
RFC 3513 променя дефинирания от IEEE идентификатор като придава важна роля на седмия бит от него [23]. Този бит се използва за различаване на глобално уникалните идентификатори от локално уникалните (уникални само в обхвата на връзката). При IPv6, ако този бит е 0 се приема, че идентификатора е локален, а ако е 1 е глобален. Остава въпроса как се определя останалата част от идентификатора и отговорът е, че това зависи от протокола на по-ниското ниво. Основните правила са следните:

- **Интерфейси с EUI-64 идентификатор –** това е най-лесния случай. Единственото изменение свързано с IPv6 е в промяната (инверсията) на седмия бит.
- **Интерфейси с MAC (Ethernet) адрес -** използва се алгоритъм за получаване на EUI-64 от 48 битов MAC. Най-общо глобалният флаг (седмия бит) на MAC адреса се обръща и се вмъква fffe между третия и четвъртия байт на MAC адреса.

Например при MAC адрес 00:8c:a0:c2:71:35, той се преобразува в идентификатор на интерфейса 028c:a0ff:fec2:7135 (Фигура 1-12).

- **Други –** при всички останали случаи идентификатора се конфигурира от мрежовия администратор.

Фигура 1-12 Превръщане на MAC адрес в EUI-64



Глобалните уникални идентификатори, част от IPv6 адреса, изглеждат като един прекрасен механизъм за глобална идентификация на абоната и реализация на мобилни услуги. Трябва обаче да се отчете, че този механизъм нарушава някои от най-важните принципи на съвременния Интернет, а именно – анонимността. С други думи е много лесно да бъде проследен даден компютър или абонат.

Проблемът е разрешен в RFC 3041 чрез позволяването на няколко идентификатора на всеки интерфейс [28]. От тях само един ще бъде глобално уникален EUI-64 идентификатор. Той ще се използва от DNS и от входящия трафик. Останалите идентификатори ще се генерират на базата на случаен принцип и ще са валидни в точно определен период от време. Тези идентификатори ще се използват от изходящия трафик. По този начин частично е разрешен проблема с проследяването и анонимността, разбира се с цената на доста усложнен механизъм и излишни идентификатори.

1.3.5.4 Anycast адреси

Anycast е адрес, който идентифицира група от възли, предоставящи една и съща услуга. Основната разлика между anycast и останалите групи адреси е, че не е обособен в отделно адресно пространство, а всъщност е най-обикновен unicast адрес. Специфичното при anycast е, че самата мрежа определя механизмите за доставката на пакети от източника до anycast адреса. Това означава, че anycast адресите се разпространяват в мрежата от маршрутизиращите протоколи и че всеки член на anycast групата е представен чрез отделен ред в таблицата на протокола. По този начин решението към кой anycast ще бъде изпратен даден пакет се взема от маршрутизращия протокол на всеки възел. От една страна това е предимство и помага при DDOS атаки (Distributed Denial Of Service), като се

атакува само най-близкия до източника на атаката възел. От друга страна е недостатък, тъй като се добавят излишни редове в маршрутизиращите таблици. Това в крайна сметка затруднява маршрутизиращия протокол. Друг недостатък на anycast е свързан с доставката на пакетите и по-точно с факта, че механизмът на доставка е оставен на IP протокола. По този начин един пакет от дадена сесия може да бъде доставен на един член на anycast групата, а друг на друг член, т.е. необходим е допълнителен локален механизъм в anycast възлите за следене състоянието на сесиите.

Последният съществен недостатък на anycast механизъма е свързан със сигурността. Няма механизъм, който да попречи на някой да се обяви за член на групата и да разпределя тази информация до източниците на трафик, чрез даден маршрутизиращ протокол.

Поради всичките тези недостатъци са въведени следните ограничения:

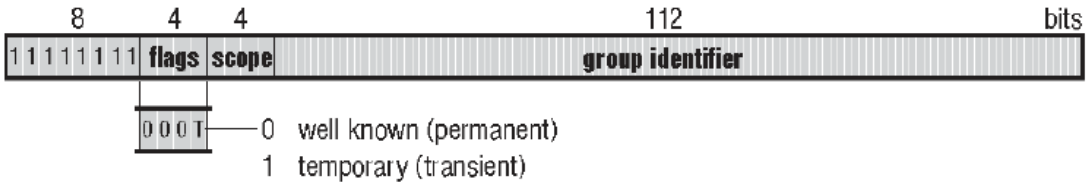
- anycast адрес никога не бива да се конфигурира на крайно устройство, т.е. такъв адрес може да се задава само на маршрутизатори;
- anycast адрес никога не бива да се използва като адрес на изпращача.

За момента anycast има ограничено приложение. Използва се основно за научни цели от университетски организации. Google DNS е една от малкото публични услуги, за които се знае, че е базирана на anycast адресация.

1.3.5.5 Multicast адреси

В сравнение с почти непознатия при IPv4 anycast адрес, multicast адресите са добре известни и широко използвани. И при двата протокола multicast е механизъм за предаване на информация от точка към много точки. При IPv6 за multicast е отделен ff00::/8/. Това означава, че IPv6 multicast адресите винаги започват с ff. Структурата на оставащите 120 бита е показана на Фигура 1-13.

Фигура 1-13 Multicast IPv6 адрес



1.3.5.6 Колко адреса има едно IPv6 устройство?

Една от най-сериозните разлики между IPv4 и IPv6 е, че при IPv4 мрежовия интерфейс има един адрес. Ако е необходимо добавянето на втори трябва да се използват виртуални под-интерфейси или да се разчита на механизми за добавяне на вторични адреси, специфични за всеки производител.

При IPv6 проблемът е решен генерално. Новата версия на протокола не само позволява повече от един адрес на интерфейс, но и прави задължителна употребата на поне няколко адреса. Адресите задължителни за всяко крайно устройство (компютър, сървър, принтер и д.р.) са:

- link-local – адрес валиден само за дадената връзка;
- unicast – публичният адрес на устройството;
- loopback address (::1) – локалния адрес на устройството, предназначен е за вътрешна комуникация;
- multicast addresses (ff01::1, ff02::1) - еквивалент на broadcast в IPv4;
- solicited node multicast address – използва се от протокола за установяване на съседство (еквивалента на ARP/RARP в IPv6);
- зададен “multicast” адрес - идентифицира групите, към които възела принадлежи.

Например компютър с един Ethernet мрежови интерфейс с MAC адрес 00:2a:0f:32:5e:d1, намиращ се в две подмрежи (2001:a:b:c::/64 и 2001:a:b:1::/64) и слушащ за multicast трафик към група ff15::1:2:3 ще получава трафик на следните интерфейси:

- fe80::22a:fff:fe32:5ed1 (link-local);
- 2001:a:b:c:22a:fff:fe32:5ed1 (първи unicast);
- 2001:a:b:1:22a:fff:fe32:5ed1 (втори unicast);
- ::1 (loopback);
- ff01::1 (използва се за комуникация към интерфейса);
- ff02::1 (всички устройства в layer 2 сегмента);
- ff02::1:ff32:5ed1 (solicited node multicast);
- ff15::1:2:3 (multicast).

Ако устройството е маршрутизатор, то трябва да поддържа освен изброените по-горе адреси и няколко anycast такива. Логично възниква въпросът измежду толкова много адреси, кой за какво трябва да се използва. RFC 3484 дефинира механизъма, чрез който се

определят адреса на изпращача и на получателя на всеки един IPv6 пакет. Основната идея на алгоритъма за избор на адрес е следната. Всяко приложение, искащо да комуникира с друго ще се опита да вземе адреса на устройството, с което трябва да комуникира. Ако получи няколко адреса, то ще опита да осъществи връзка с първия. Ако това не стане, ще използва следващия и така нататък. Другият основен принцип е, че при избор на адрес или посока в дадена мрежа винаги се взема префикса с най-много цифри, отговарящи на зададения краен адрес.

1.3.6 Протоколи за откриване на съседни IPv6 устройства

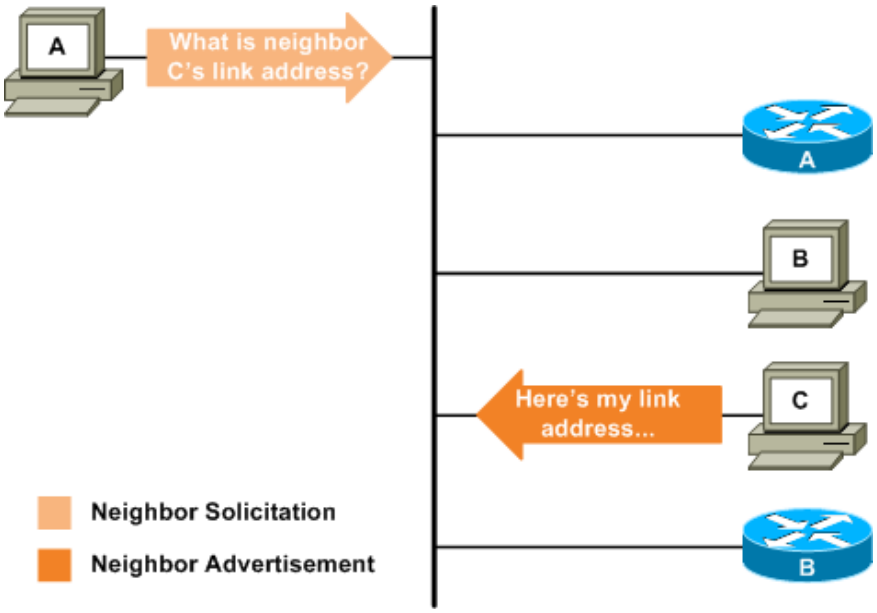
Според [29] NDP (Neighbor Discovery Protocol) може да бъде определен като сбор от механизми, използвани от IPv6 устройствата за различни видове локални операции за слоя на данните. Част от механизмите в NDP присъстват и в IPv4, а други се появяват за пръв път при IPv6. Примери за представители на първата група са RDISC (Router Discovery), ARP (Address Resolution Protocol) и ICMPv4 (Internet Control Message Protocol) версия 4, а на втората - механизъма за "stateless" конфигурация на IPv6 адреса.

Важно е да се спомене, че NDP се появява в последствие и първоначално механизмите са били описани в отделни RFC препоръки.

Комбинацията от тези протоколи позволява на IPv6 устройствата да откриват други такива, налични в layer 2 сегмента, включително и наличните маршрутизатори. От съобщенията, изпратени от маршрутизаторите, останалите възли могат да конфигурират сами адресите си. Алгоритъмът също така има и вградени механизми за откриване на дублирани адреси и откриване на отпаднали възли.

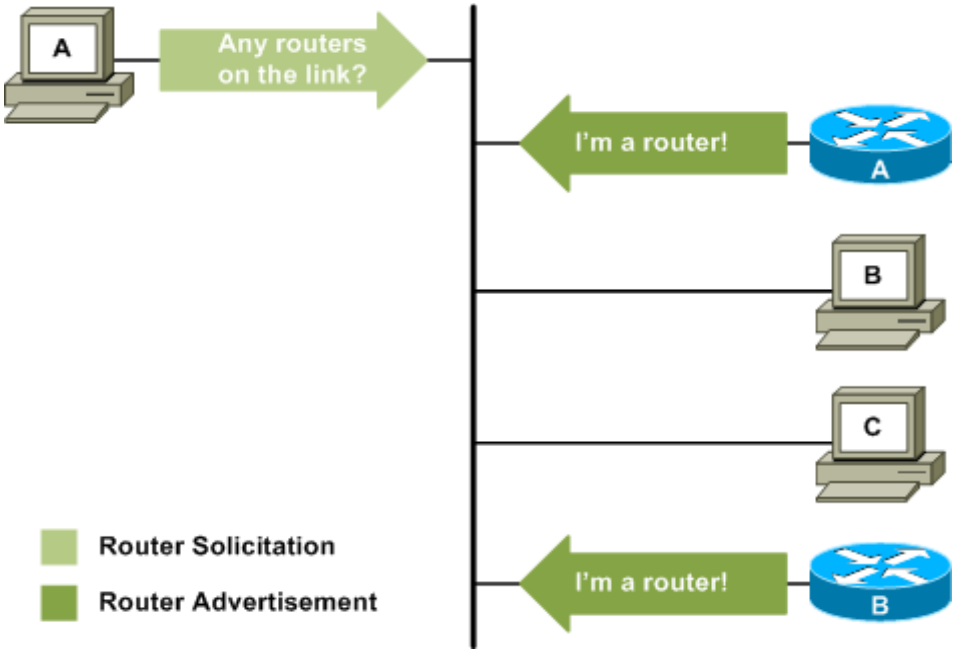
На Фигура 1-14 е показан процеса по определяне на локалния адрес на даден възел, намиращ се в същия layer 2 мрежови сегмент. За целта са използвани две ICMPv6 съобщения. Neighbor Solicitation (type 135) се изпраща по multicast от възела, търсещ адреса на даден съсед, а съседния възел отговаря със своя local-link адрес чрез Neighbor Advertisement (type 136) ICMPv6 съобщение [30].

Фигура 1-14 IPv6 Neighbor address resolution



На Фигура 1-15 е демонстриран механизъма използван в IPv6 за разкриване на съседните устройства в даден layer 2 сегмент. Съобщенията се изпращат или на „multicast“ адрес „all nodes”, или като отговор на „router solistication” съобщения, изпратени от дадено IPv6 крайно устройството. Съобщенията, които обявяват даден маршрутизатор може да съдържат множество префикси. Те се използват за „stateless“ конфигурация на IP адресите на устройствата в сегмента, за поддръжка на списък с префикси и за избягване на дублирани IP адреси [31]. Възлите използват списъка и за маршрутизация. По този начин крайните възли знаят към кой маршрутизатор да изпратят даден пакет, а маршрутизаторите знаят към кой сегмент да насочат дадени пакети.

Фигура 1-15 Механизъм за разкриване на маршрутизаторите в даден L2 сегмент

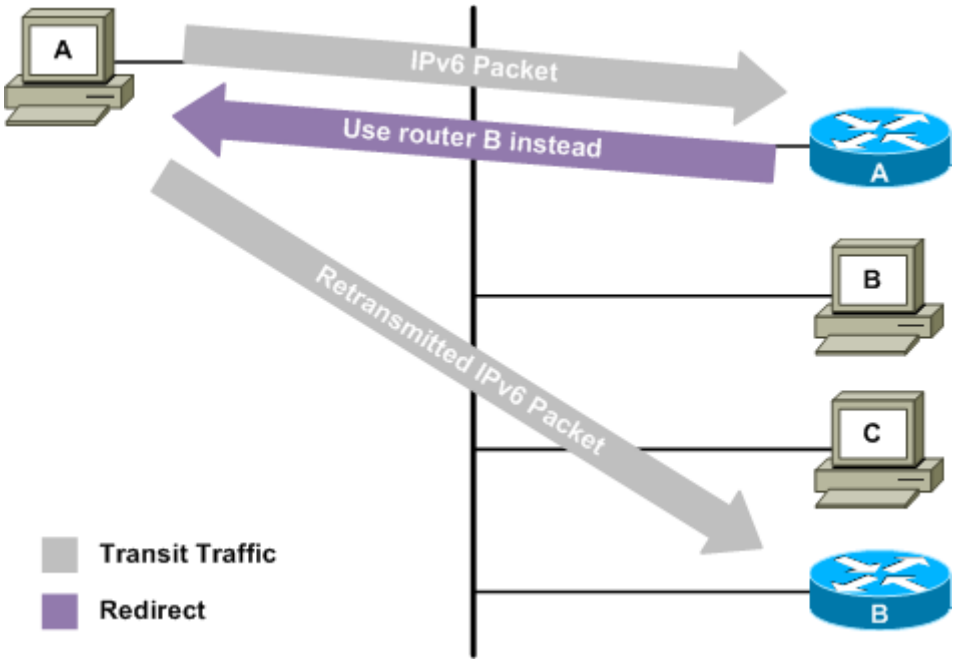


Съобщенията за сигнализация съдържат следните полета:

- префикси – това може да са префикс, дължина, период на валидност, флагове, показващи дали префиксът е валиден за дадения сегмент и дали може да се използва за автоматична конфигурация;
- информация за маршрутизатора, предлагащ път по подразбиране;
- допълнителна информация за устройствата като брой възли, максимален размер на пакетите за сегмента.

На Фигура 1-16 е демонстрирано пренасочване на IPv6 към коректния маршрутизатор. Механизмът е подобен на използвания при IPv4 ICMP redirect. В случая се използва ICMPv6 Redirect (type 137) съобщение от маршрутизатора, към който пакета първоначално е насочен. Веднъж получено това съобщение от източника на трафик, той пренасочва потока от пакети към коректния маршрутизатор.

Фигура 1-16 Пренасочване на IPv6 пакети в L2 сегмент



1.4 Автоматизиране на процеса на конфигурация на IPv6 адреси

IP адресите може да бъдат конфигурирани ръчно или да бъдат зададени автоматично. Това са двете основни алтернативи, независимо дали става дума за IPv4 или за IPv6. При IPv4 автоматичната конфигурация на адреса на устройството може да стане по DHCP (Dynamic Host Configuration Protocol) или чрез PPP (Point to Point Protocol). При IPv6 адресът може да бъде зададен автоматизирано чрез следене на състоянието (по DHCPv6 [32] или по PPP протокол [33]) или “stateless”, т.е. без да се следи състоянието от всеки един локален IPv6 маршрутизатор.

В общия случай ръчната конфигурацията на IP адреси е най-често срещаната и винаги възможна опция. Тя е приложима в малки до средни мрежови инфраструктури, при които контролът на процеса на задаване на адресите е изключително важен. Тази опция е трудно осъществима в големи IP мрежи каквито са мрежите на съвременните доставчици [29].

В първият случай IP адресът на крайното устройство ще бъде зададен по DHCP протокол. Протоколът е реализиран чрез клиент в крайното устройство и сървър. Сървърът съдържа информация за множеството възможни IP адреси, които могат да бъдат конфигурирани на крайните клиентски устройства. Освен статичната информация за

множеството от адреси, сървърът съдържа и информация за състоянието на всеки адрес - кои адреси вече са конфигурирани и кои са все още свободни.

PPP е протокол за изграждане на връзки от точка до точка върху различни L1/2 среди. Основното предимство на PPP е, че позволява интеграция на процеса по автентикация на клиента, изграждане на сесията и задаване на IP адрес. След като клиентът удостовери своята самоличност и бъде изградена сесия на слой 2, PPP използва NCP, за да конфигурира IP адреса на крайното устройство. В най-често срещания случай PPP сървърът използва RADIUS (Remote Authentication Dial In User Service) за удостоверяване на самоличността и за задаване на адреса, който да бъде конфигуриран на крайното устройство.

PPP заедно с RADIUS и DHCP са протоколи, способни да конфигурират автоматизирано и динамично IPv4/ IPv6 адреси в голям мащаб. Те са подходящи за инфраструктурата и услугите предоставяни от съвременните доставчици на мрежови услуги. IPv6 позволява и трети вариант за автоматична конфигурация на адреса на крайното устройство без следене на състоянието (stateless) и базирана на префиксите, обявявани от локалните маршрутизатори. Предимствата на IPv6 SLAAC (Stateless Address Auto-Configuration) [34] са:

- Не се изисква никаква предварителна конфигурация на крайните устройства;
- Не се изисква наличието на отделен „сървър“, който да следи състоянието и да задава IPv6 адресите;
- Не се изисква почти никаква конфигурация на маршрутизаторите.

SLAAC се базира на механизъм за откриване на маршрутизаторите и предоставя „plug-and-play” свързаност в две фази. Във фаза 1 се задава локалния IPv6 адрес, а във фаза 2 глобалния.

Фаза 1 - Генериране на локален IPv6 адрес

- Всеки път когато бъде пуснат даден IPv6 интерфейс се генерира локален IPv6 идентификатор на интерфейса от мрежа FE80::/10.
- Веднъж генерирал адреса, възелът е длъжен да провери дали адреса е уникален за дадения локален сегмент. Това става чрез изпращането на

съобщение „neighbor solicitation“ с адрес на получателя равен на генерирания адрес. Ако възелът получи отговор, тогава адресът е дублиран и процесът спира (изисква намеса от оператор).

- Ако адресът е уникален, възелът го задава като локален адрес на интерфейса, за който е бил генериран.

В този момент възелът има IPv6 свързаност към всички други възли, намиращи се в същия сегмент. Ако възелът изпълнява ролята на мрежово устройство, той не продължава към фаза 2 и глобалният адрес трябва да бъде конфигуриран по някакъв друг начин. Ако възелът изпълнява роля на крайно устройство, тогава започва фаза 2.

Фаза 2 - Задаване на глобален IPv6 адрес

- Процесът съвпада с показания на Фигура 1-15 и започва с изпращането на „router-solistication” съобщение от възела, търсещ глобален адрес, към всички локални маршрутизатори. Те от своя страна отговарят със съобщения “router-advertisements”. Ако маршрутизаторът поддържа SLAAC, съобщението “router-advertisement” ще съдържа префикса на мрежата, която да бъде използвана за целта.
- Веднъж получил префикс от маршрутизатор, търсещият генерира глобален адрес чрез прибавяне на своя интерфейс идентификатор към получения префикс.
- Следва проверката за дублиран IPv6 адрес. Процесът протича по същия начин както и във фаза 1.
- Ако адресът не е дублиран, крайното устройство го задава на интерфейса и с това процесът приключва.

Подходът адресите да бъдат задавани, без да се следи състоянието на адреса от централизиран възел, е подходящ в случаите, когато адресът трябва да е уникален и маршрутизируем и да бъде зададен на минимална цена (например с минимален разход на процесорно време и памет). От друга страна DHCPv6 изисква повече ресурси и е по-подходящ, в случаите когато е необходим по-стриктен контрол и ръчната конфигурация не е възможна [32]. Важно е да се отбележи, че при IPv6 нищо не пречи да се използват и

двата подхода едновременно и крайното устройство да определя кой от всичките си адреси да използва.

1.5 Механизми за преход от IPv4 към IPv6

Скоро след създаването на IPv6 и дефинирането му в RFC 2460 са започнали да се появяват и първите механизми за преход от IPv4 към IPv6. До момента са излезли повече от десет RFC документи, описващи механизмите за преход и поне още десет, описващи как те да бъдат комбинирани едни с други в един или друг случай. Най-общо може да ги обединим в три групи [8], [9], [35], [36], [37]: двоен IP стек, механизми, базирани на превод от IPv4 към IPv6 и механизми, базирани на изграждане на тунели.

Двойният IP стек се основава на едновременната поддръжка на IPv4 и IPv6. Разгледан е класическия двоен IP стек и олекотената форма – DSLite (Dual Stack – Lite) [38].

Механизмите за превод се основават на транслацията на адреси и пакети от IPv4 към IPv6. Разгледани са NAT-PT [39], NAT64/NAT46 [26]и CGN (Carrier Grade NAT) [40].

Изграждането на тунели се използва за пренос на IPv6 трафик през IPv4 домейн и обратно. Разгледани са подробно 6in4 6to4, 6rd, 4over6. Тунелите могат да се изграждат автоматично или на базата на статична конфигурация. Всеки един от тези два подхода има своите предимства и недостатъци. Статичната конфигурация е по-добра от гледна точка управлението на мрежата и по-специално от гледна точка на сигурността (все пак е известно между кои две точки има тунел). Автоматичните тунели също зависят от предварителната конфигурация, но след като тя бъде направена се образуват тунели между множество от крайни или междинни точки на база на текущия трафик. Съответно при тях контролът е по-труден и нивото на сигурност е занижено.

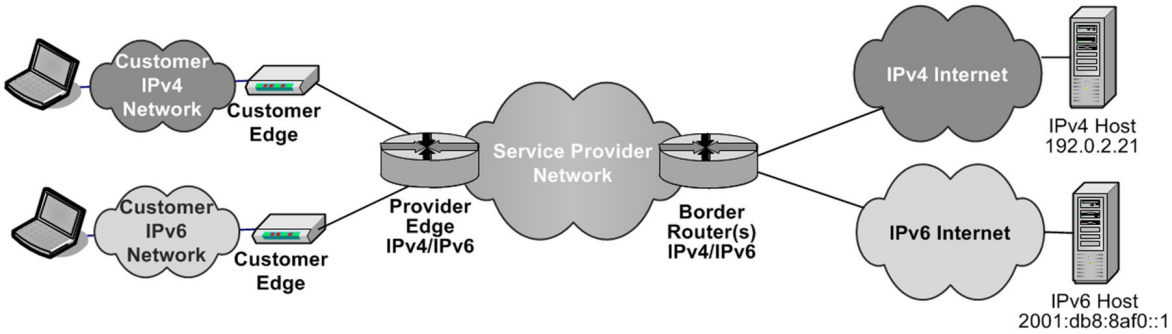
Тук са разгледани и две от технологиите за внедряване на IPv6 в IPv4 MPLS инфраструктура на Интернет доставчик – 6PE (IPv6 Provider Edge) и 6VPE [7].

1.5.1 Класически двоен IP стек

Двойният IP стек се изразява в едновременната поддръжка на IPv4 и на IPv6 в мрежата (Фигура 1-17). Най-доброто и пълно описание е дадено в RFC 4213 [37]. Препоръката специфицира два основни механизъма за преход от IPv4 към IPv6 –

изграждане на тунели и двоен IP стек. Едновременната поддръжка на двата IP стека ще реши множество проблеми, свързани с прехода от IPv4 към IPv6. Например ще позволи едно и също оборудване да изпълнява функции както в IPv4, така и в IPv6 среда. Това е технологията, която реално ще допринесе най-много за прехода от IPv4 към IPv6. Двойният IP стек е задължителното условие за използването на някои от останалите механизми. Например, няма как да бъде направен 6to4 или 4to6 тунел, ако двата края на тунела не са предварително конфигурирани с двоен IP стек. Същото важи и за механизмите за преобразуване на адреси 4to6 и 6to4. Те не биха били възможни, ако възелът, който извършва транслацията не е предварително конфигуриран да поддържа двоен IP стек.

Фигура 1-17 Dual stack (двоен IP стек)



Наред с безспорните си предимства двойният IP стек има и редица недостатъци. На първо място наличието на двоен IP стек из цялата мрежа предполага конфигурацията, наблюдението, управлението и защитата не на един, а на два IP стека. Това удвоява „сложността“ на мрежата, създава възможност за допускане на грешки и за пробиви в сигурността. Поради тези причини добрата практика е да се ограничи масовата употреба на двоен IP стек и същият да се използва само там където е необходимо - например в граничните възли между IPv4 и IPv6 на устройствата, отговорни за NAT64 или в крайните точки на 6to4/ 4to6 тунели.

1.5.2 Олекотен двоен стек (Dual-Stack DS-lite)

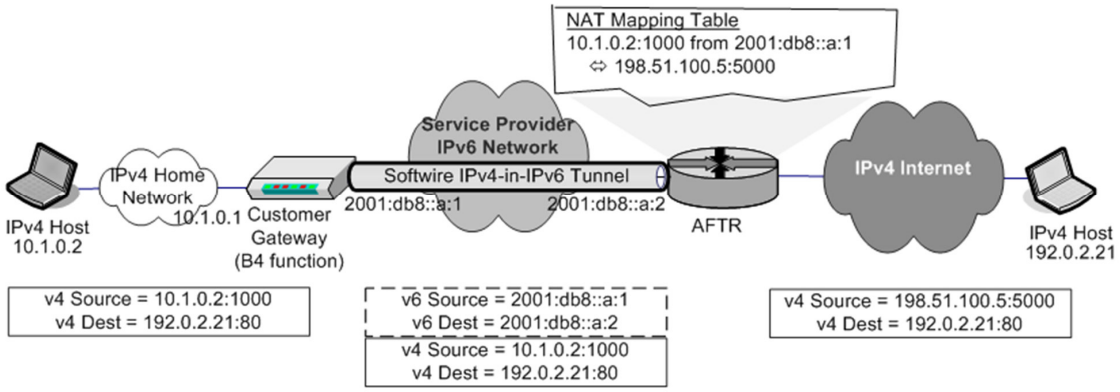
Олекотеният двоен IP стек е технология, позволяваща на доставчик да мигрира опорната си мрежа към IPv6 и да продължи да използва досегашната IPv4 адресация, зададена на крайните клиентски устройства [38]. В най-общия случай доставчикът задава

IPv4 адрес на клиентското устройство. То от своя страна изпълнява функциите на DHCP сървър и раздава адреси на устройствата, намиращи се зад него. DS-lite е технология, която може да се използва за раздаване на IPv4 адреси на крайни клиентски устройства, които не поддържат IPv6 като в същото време основната инфраструктура е IPv6 базирана.

DS-lite е базиран на:

- B4 елемент (Basic Bridging BroadBand) – предава трафик от домашна IPv4 мрежа към IPv6 като изгражда Softwire IPv4-върху-IPv6 тунел. B4 може да бъде реализиран на клиентското устройство CG (Customer Gateway) или в мрежата на доставчика.
- Softwire IPv4-over-IPv6 тунел – предава IPv4 трафика между B4 и AFTR (Address Family Translation Router) в 4over6 тунел. Тунелът се изгражда между началната B4 и крайната AFTR точка.
- AFTR – крайна точка на “softwire” тунела и изпълнява NAT44.

Фигура 1-18 Dual stack lite (олекотен двоен IP стек)



На Фигура 1-18 е илюстрирана архитектурата на DS-Lite и е показана взаимовръзката между трите компонента. Най-вляво IPv4 възел получава адрес 10.1.0.2 от CG по DHCP. Прави се допускането, че 10.1.0.2 се опитва да се свърже с web страница, чийто адрес е 192.0.2.21. Това предполага сесия с параметри IPv4 Source Address 10.1.0.2:15535 и IPv4 Destination Address 192.0.2.21:80. В случая се използва сокет IP адрес и номер на порт. Възелът препраща пакетите от сесията според своя маршрут по подразбиране (default route) към CG устройството, изпълняващо функция B4. То изгражда

IPv4inIPv6 тунел към устройството, изпълняващо функция AFTR. AFTR играе ролята на крайна точка на тунела и прави NAT44. Преобразуването на адреси се изразява в заместване на частния IPv4 адрес [41] с публичен. Това предполага наличието на множество (pool) публични IP адреси на AFTR устройството, които да бъдат споделени между голям брой DS-Lite клиенти. След преобразуването сокетът 10.1.0.2:1000 е заменен с 198.51.100.5:5000.

При трафик в обратна посока, от 192.0.2.21 към 10.1.0.2, AFTR следи не само състоянието на транслираните сесии, но и кой DS-lite тунел е асоцииран с дадената сесия. В обратна посока AFTR енкапсулира IPv4 в IPv6 и го насочва в тунела към 2001:db8::a:1. B4 декапсулира пакетите и ги препредава към 10.1.0.2.

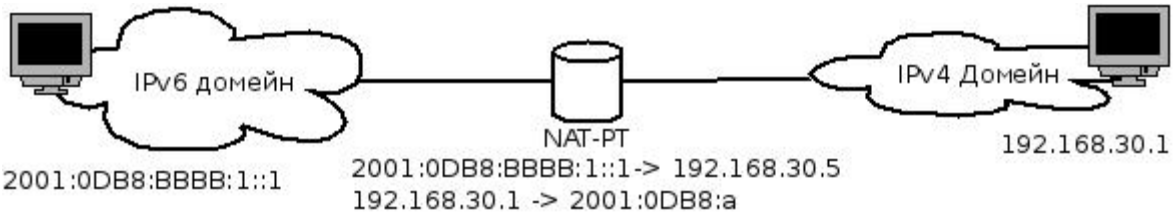
DS-Lite е механизъм, подходящ за доставчици с голям брой IPv4 клиенти, предвиждащи широкомащабни миграции към IPv6. При подобна миграция винаги ще има устройства, които няма да поддържат IPv6. Чрез DS-lite тези устройства няма да загубят свързаност и ще продължат да работят и комуникират по IPv4 върху IPv6 с останалия IPv4 свят.

1.5.3 Network Address Translation/Protocol Translation (NAT-PT)

NAT-PT е един от първите механизми за преобразуване на адреси и портове между IPv4 и IPv6. Той е дефиниран в RFC 2766 [39] и е директен наследник на NAT и PAT (Port Address Translation) механизмите за превод на адреси и портове в IPv4. Основното му предназначение е да позволи двустранна, комуникация между IPv4 и IPv6 мрежови устройства. NAT-PT има две разновидности – статична и динамична:

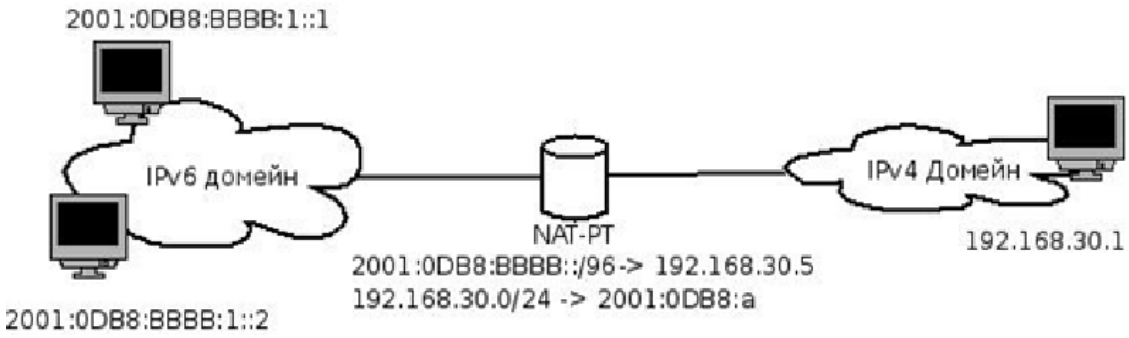
- Статичен NAT - директно преобразуване на точно определен IPv4 адрес към точно определен IPv6 адрес и обратно (Фигура 1-19) или на комбинация от точно определени IPv4 адрес и порт към друга комбинация от точно определени IPv6 адрес и порт.

Фигура 1-19 Статичен NAT-PT



- Динамичен NAT - преобразуване на по-голямо множество от IP адреси към по-малко. Тази част на механизъма е директен наследник на Port overload механизъма в IPv4, т.е. сесиите генерирани от множество IPv6 хостове се транслират към един или няколко IPv4 такива, като освен адреса се преобразува и порта на източника на трафик (Фигура 1-20). Динамичният NAT-PT е подходящ за случаите, когато не са предварително известни двойките IP адреси, които ще комуникират между двата домейна. Той е подходящ също и в случаите, когато трябва да се спестят IPv4 адреси. Разбира се динамичния NAT-PT е трудно приложим в голям мащаб поради огромния брой сесии, генерирани от съвременните приложения. Само за пример Skype или торент програма генерират повече от 100 едновременни сесии.

Фигура 1-20 Динамичен NAT-PT



Предимства и недостатъци на NAT-PT според RFC 4966 [42]

Предимства:

- Директен наследник на съществуващите механизми за превод на адреси.
- Лесен за имплементация и конфигурация.

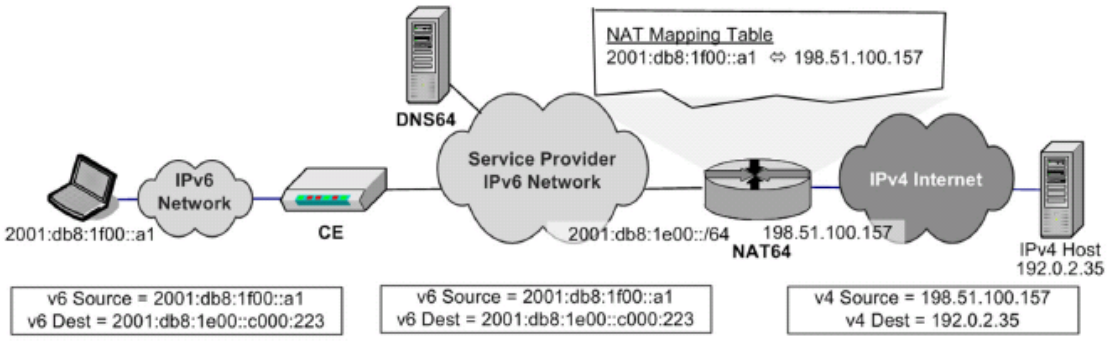
Недостатъци:

- При използване на статичен NAT са необходими толкова IPv4 адреси колкото и IPv6. Това означава, че няма друго предимство освен спестяване на IPv4 адресно пространство.
- При използване на динамичен NAT се скрива адресът на крайното устройство и системите за регламентиран според закона достъп за подслушване на мрежовия трафик не могат да идентифицират източника.
- Не се справя добре в ситуации, при които има вградени IP адреси в полезната част на пакетите. Типичен пример за това е DNS и протоколите за VOIP (Voice over IP). Проблемът може да бъде избегнат, ако към механизъма за превод бъде добавен и специализиран приложен шлюз, който да заменя адресите на ниво полезна информация. Това предполага разчитане на информацията в приложния слой, което изисква значително количество изчислителни ресурси. Голяма част от трафика в една съвременна телекомуникационна мрежа е именно такъв и съответно този недостатък е една от основните причини IPv4 NAT, а също и NAT-PT, да са трудно приложим в голям (Интернет) мащаб.
- Предизвиква нежелани прекъсвания на сесиите при протоколи с вградени механизми за запазване на връзката (keep-alive).
- Не работи за трафик от точка до много точки (multicast).
- Не се справя добре, ако се налага пренасочване на трафика (redirect).
- Не работи, ако се налага мобилност.

1.5.4 NAT64/ NAT46 и DNS64/DNS46

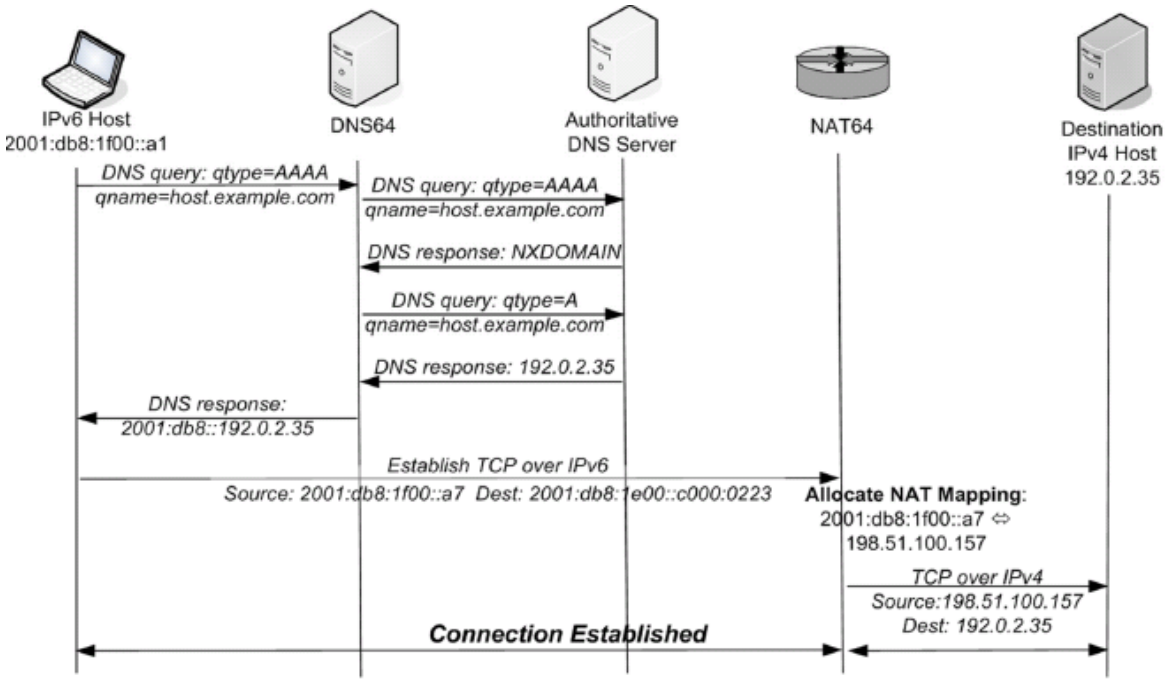
NAT64/46 в комбинация с DNS64/46 са група механизми за преобразуване на мрежови адреси от IPv6 към IPv4 и обратно [26]. Групата е известна като NAT64/DNS64. Основната цел е да се осигури прозрачна комуникация между двата домейна и да бъдат избегнати недостатъците на NAT-PT.

Фигура 1-21 NAT-64



На Фигура 1-21 и Фигура 1-22 е демонстрирана комбинацията от NAT64/DNS64. NAT64 е механизъм за преобразуване на адреси от IPv6 към IPv4. Той се прилага в стратегии, при които мрежата е изцяло мигрирана към IPv6. Единствено възлите с NAT64 са с двоен IP стек и осъществяват връзката между двата домейна. NAT64 зависи от допълнителен механизъм - DNS64. Това е допълнителна функционалност на съществуващите DNS сървъри, която изпълнява специални рекурсивни заявки. DNS64 отговаря нормално на валидните IPv6 AAAA заявки (Фигура 1-22). Ако за дадена заявка не получи AAAA запис, DNS64 генерира допълнителна заявка за IPv4 A запис. Ако получи отговор, преобразува IPv4 адреса във валиден AAAA отговор (IPv6 адрес). За целта конструира нов IPv6 адрес, като прибавя /96 префикс пред IPv4 адреса на хоста местоназначение. Този префикс, дошъл от A записа, се отстранява от NAT64 възела. По този начин се получава 128 битов адрес като последните 32 бита идват от IPv4 адреса на A записа. По този начин трафикът от крайната точка ще бъде насочен към NAT64 възела, който ще отстрани добавения префикс и ще препрати заявката към IPv4 домейна.

Фигура 1-22 NAT64/DNS64 сигнализационен поток



NAT64 поддържа два режима на работа: „stateless" и „statefull".

„Stateless" NAT64 [43] е механизъм за преобразуване на IP и ICMP адреси на пакети от IPv6 към IPv4 и обратно, който преобразува адресите без да следи за състоянието на сесиите, към които тези пакети принадлежат. Този подход е значително по-евтин по отношение на ресурси от "stateful" подхода, но има един съществен недостатък. За преобразуването са необходими точно толкова IPv4 адреси колкото и IPv6. Механизмът не е подходящ за оператори, които биха искали да спестят "IPv4" адресно пространство и е подходящ за такива, които въпреки, че в момента все още имат достатъчно IPv4 адреси, мислят за бъдещето и предпочитат да извършат прехода "навреме".

„Stateful" NAT64 [44] е механизъм за превод от IPv6 към IPv4 и обратно, който преобразува пакетите и следи за състоянието на сесиите, към които тези пакети принадлежат. Съответно той позволява преобразуването на много IPv6 адреси към малко IPv4 такива. Механизмът е подходящ за оператори, които не са извършили прехода на време и са в ситуация с почти изчерпано IPv4 адресно пространство. Недостатък на "statefull" подхода е необходимостта от много по-големи системни ресурси за следене на състоянието на отделните сесии.

Предимства и недостатъците на NAT64

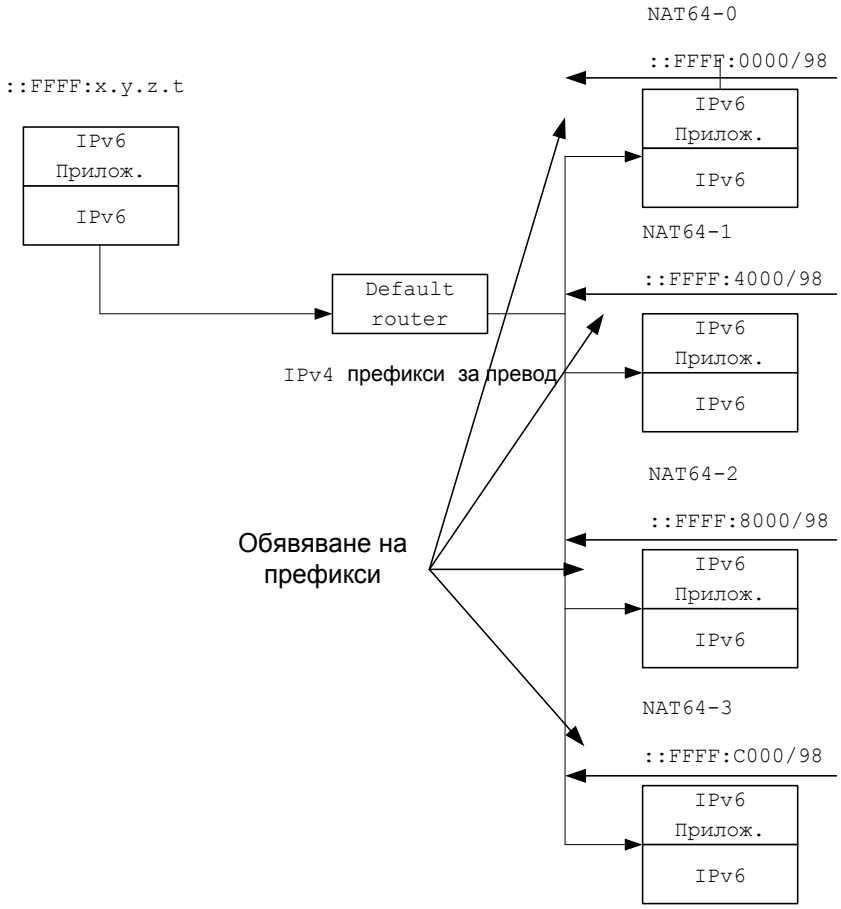
Недостатъци:

- NAT64 зависи от състоянието на маршрутизиращата информация. Ако тя е компрометирана, преобразуваните пакети може да не достигнат до коректния NAT64 възел и да се получат неочаквани последствия, като препредаване до изтичане на TTL записа или “тихомълком” да бъдат отстранени.
- Недостатъците характерни за традиционното преобразуване на адреси (RFC 2993) [45].
- Stateless NAT64 би бил по-трудно приложим за фиксирани оператори с голям брой IPv4 клиенти и малко останали свободни IPv4 адреси.
- Необходима е манипулация на DNS записите. Това означава добавяне на допълнителен механизъм, който се базира на допълнителни заявки и съответно генерира допълнително забавяне.

Предимства:

- Гъвкавост – комбинацията от NAT64 и DNS64 може да бъде изключително гъвкава по отношение на разпределението на капацитета, който да преминава през NAT64 устройствата. Това се постига чрез добавяне на допълни префикси в DNS64, допълнителни NAT64 устройства и насочване на трафика към всяко едно от тях (Фигура 1-23). Основният NAT64 префикс е разделен на четири подмножества, конфигурирани на четири отделни устройства. Всяко едно от тях обявява в мрежата зададения му NAT64 префикс. DNS64 може да бъде конфигуриран да работи със всяко едно от устройствата и дори да балансира трафика между тях.

Фигура 1-23 Гъвкавост при NAT64



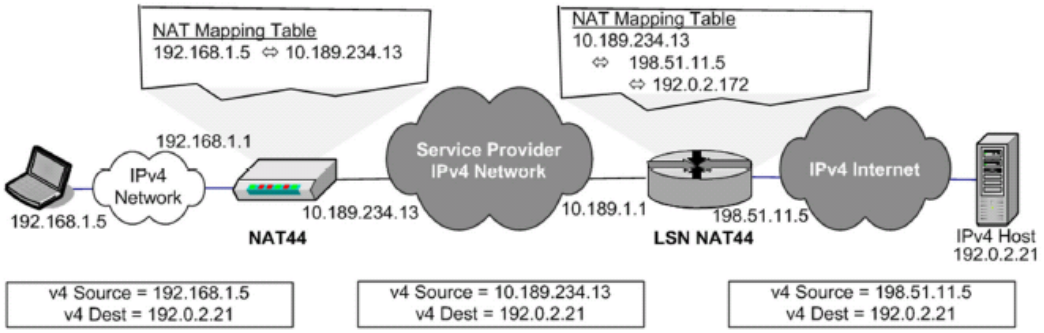
- Stateless NAT64 е много подходящ за нови оператори с IPv6 базирани мрежи или за мобилни оператори, които така или иначе и в момента имат NAT между вътрешната мрежа с пакетна комутация (2G, 3G, LTE) и Интернет пространството [36].
- Не се налага фрагментация или увеличаване размера на пакета (все недостатъци на технологиите, които използват изграждане на тунели).

1.5.5 Carrier Grade NAT (NAT444)

Класическият Carrier Grade NAT444 е механизъм за двойно преобразуване на адреси, предоставящ възможност на доставчиците на услуги да отложат прехода към IPv6 за известен период от време [40]. Механизмът се основава на възел (Large Scale NAT - LSN 444) за преобразуване на частни IPv4 адреси към публични IPv4 такива в изключително голям мащаб. Целта е операторът да споделя едни и същи публични IPv4 адреси измежду

голям брой абонати. Механизмът удължава „живота“ на съществуващите IPv4 мрежи и CE устройства с цената на ограничаване на броя сесии, които клиентите потенциално могат да се опитат да изградят. NAT444 има и други негативни ефекти, например „скрива“ потребителите и прави невъзможна точната им локализацията по IP адрес. По този начин възпрепятства работата на системите за регламентирано според закона подслушване на мрежови трафик и усложнява откриването на източника на конкретен вид трафик.

Фигура 1-24 NAT444



На Фигура 1-24 е демонстриран NAT444. Сценарият се състои от два IPv4 възела, комуникиращи помежду си и две NAT44 устройства между тях. Първият NAT44 се извършва от CE като преобразува частното IPv4 адресно пространство на клиента към IPv4 адресно пространство, предоставено от доставчика. Вторият NAT44 се извършва от LSN 444 и преобразува IPv4 адресите на доставчика към публично IPv4 адресно пространство. Това става чрез преобразуване не само на адреса, но и на порта на източника на пакета. Последното преобразуване е динамичен IPv4 NAT и PAT.

NAT444 е по-скоро стратегия за преобразуване на адреси в глобален мащаб, отколкото механизъм за преход към IPv6. Първоначалната стратегия позволява промени, например подмяна на втория NAT44 с NAT46. По този начин доставчикът може да мигрира своята мрежа към IPv6 без съществуващите крайни клиенти да разберат за това.

Предимства и недостатъци на NAT444

Предимства:

- Не се налага подмяна или повторна конфигурация на крайното клиентското оборудване.

- Не се налага манипулация на DNS записите (DNS 46/64).
- Не се налага фрагментация или увеличаване размера на пакета (все недостатъци на технологиите, които използват изграждане на тунели).
- Във вариант NAT464, CGN може да се използва за преход към IPv6 без крайните клиенти да разберат това.

Недостатъци:

- Ограничен брой сесии на клиентско устройство. Това би довело до потенциални проблеми при работа с приложения, базирани на голям брой сесии (AJAX, SKYPE, Facebook, RSS, Torrents, Emule).
- Необходимост от специално внимание при работа с VOIP приложения.
- Механизмът не е гъвкав по-отношение на капацитета. Това означава, че няма възможност за плавно увеличение на броя сесии, с които мрежата може да обработи.
- Механизмът във вариант NAT444 не предлага вариант за преход към IPv6.

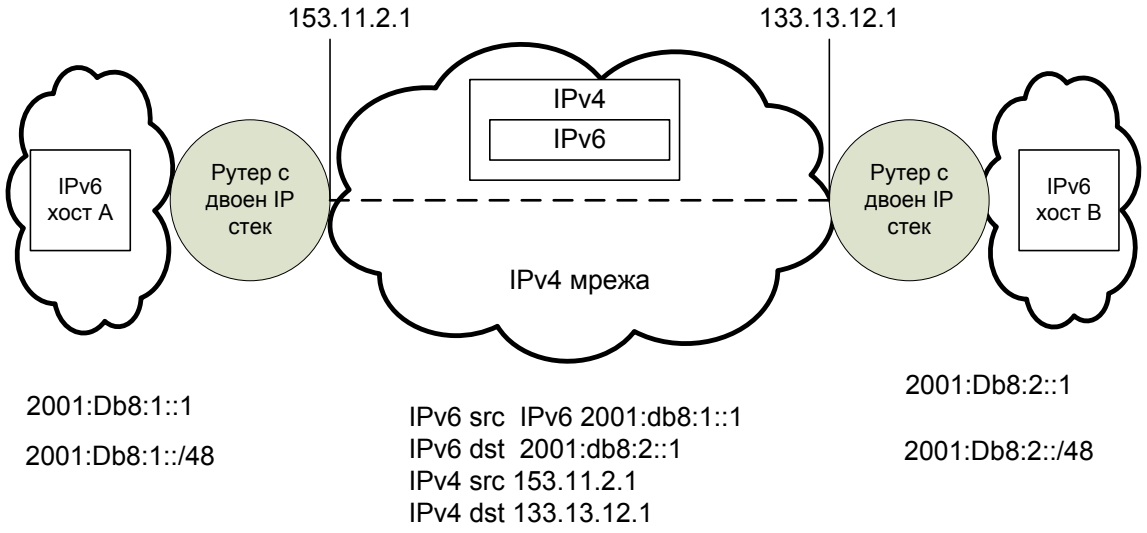
Макар да не е директен механизъм за преход към IPv6, NAT444 има едно ключово предимство - предлага подход, който ако бъде допълнен с допълнителни механизми за преход, позволява плавна и ниско рискова миграция на съществуващите IPv4 инфраструктури към IPv6. Допълнителните механизми могат да бъдат двоен IP стек, 6rd, 6CE, 6PE, NAT64 и др.

1.5.6 6in4

6in4 (Фигура 1-25) е статичен механизъм за свързване на отдалечени IPv6 зони през IPv4 среда [44]. Механизмът използва изграждане на тунели между статично конфигурирани крайни точки [37]. 6in4 работи на базата на изграждането на IPinIP тунели, като предаваните IPv6 пакети са със стойност на полето протокол - 41. Тази стойност е специално заделена за енкапсулиран IPv6 трафик и се използва и от други механизми като 6to4, 6rd и 6over4. При 6in4, а и при останалите механизми, използващи IPinIP тунели, IPv6 заглавната част следва директно след тази на IPv4. Следователно, ако максимално големият пакет, който може да бъде пренесен през дадена IPv4 среда, е 1500 байта, то

максимално големият IPv6 пакет, който може да бъде пренесен без да бъде фрагментиран, е 1480 байта (1500 – 20 байта IPv4 заглавна информация).

Фигура 1-25 6in4 – Принципна схема



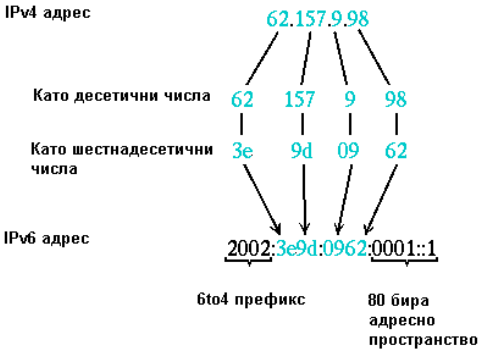
Предимства и недостатъци:

- 6in4 не предвижда механизми за защита. Ако бъде откраднат (spoofed) IPv4 адреса на тунела, може да бъде генериран произволен IPv6 трафик, който да стигне до крайните точки.
- Като предимство може да се тълкува факта, че механизмът е статичен. По този начин все пак предоставя контрол и може да бъде минимизиран ефекта от горния недостатък.
- Статичността не винаги е предимство. Статичната конфигурация на двете крайни точки на тунела може да се превърне в недостатък при голям брой тунели.
- Механизмът не специфицира как се процедира, ако междинна защитна стена блокира IP пакети със стойност на полето протокол 41 или ако пакетите биват преобразувани от NAT механизъм.

1.5.7 6to4

6to4 е специфициран в [46] и представлява механизъм за пренос на IPv6 трафик върху IPv4 среда, чрез автоматичното изграждане на 6to4 IPinIP тунели между IPv6 възлите (Фигура 1-26, Фигура 1-27). 6to4 пакетите също са със стойност 41 на полето протокол в заглавната част на IPv4 пакета. Механизмът предвижда уникален IPv6 префикс за всеки, който вече има IPv4 публичен адрес. Уникалността на префикса се постига чрез преобразуване на IPv4 адреса на шлюза в IPv6 адрес чрез конвертиране в шестнадесетичен формат и добавяне към префикс 2002::/16.

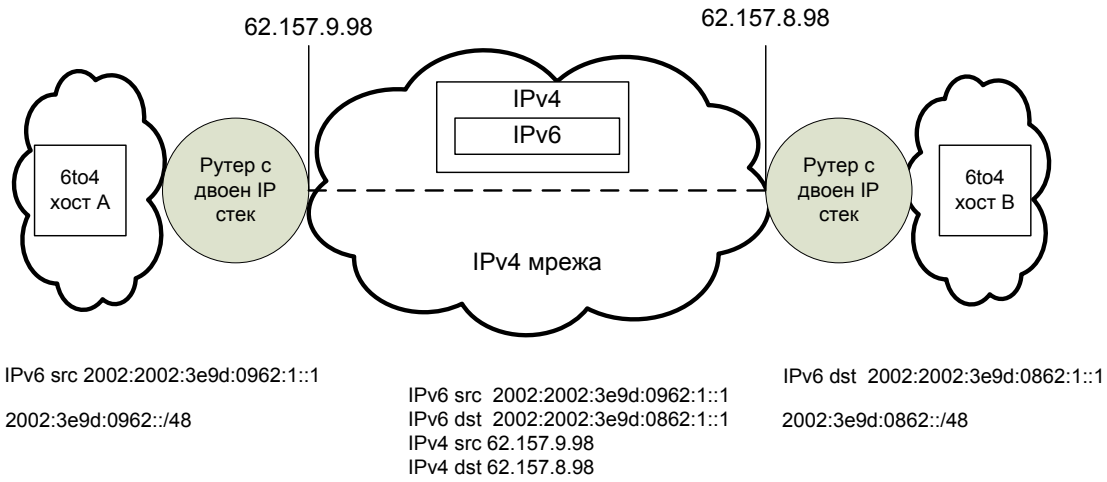
Фигура 1-26 Получаване на 6to4 адрес



По този начин всяка една от изолираните IPv6 зони получава префикс във формат 2002:V4ADDR::/48, където V4ADDR е IPv4 адрес за идентификация на 6to4 шлюза и зоната зад него.

Крайните устройства получават адреси от това адресно пространство и маршрут по подразбиране от 6to4 шлюза.

Фигура 1-27 6to4 Начин на работа



Още от самото си начало 6to4 има редица недостатъци описани в [46]:

- Стандартът е уязвим от гледна точка на сигурност и е идеално средство за разпространение на анонимни атаки от единия домейн към другия.
- Не позволява използването на частни адреси за изграждането на тунела. Това автоматично означава, че той е трудно приложим в частни мрежи. Например между офисите на дадена организация, наемаща свързаност помежду им от доставчик.
- Механизмът не специфицира как се процедира, ако междинна защитна стена блокира IP пакети със стойност на полето протокол 41.
- Комбинацията от предните два недостатъка може да се обобщи като невъзможност за работа в NAT среда. Пример за това са случаите, в които устройство, искащо да се свърже с IPv6 е с частен адрес и е зад IPv4 NAT. Този недостатък може да се избегне, ако 6to4 и NAT са на едно устройство, но това би означавало добавяне на 6to4 функционалност към всеки един съществуващ NAT, което е трудно постижимо.

1.5.7.1 6rd (IPv6 Rapid Deployment)

6rd е механизъм за „скоростно въвеждане" на IPv6 върху IPv4 мрежови инфраструктури. 6rd е директен наследник на 6to4 като използва приблизително същата схема и избягва част от архитектурните му недостатъци. Механизмът е стандартизиран в [47] и предоставя възможност на доставчиците на услуги да предоставят IPv6 адрес и свързаност на крайните си клиенти върху съществуващата IPv4 инфраструктура. Методът

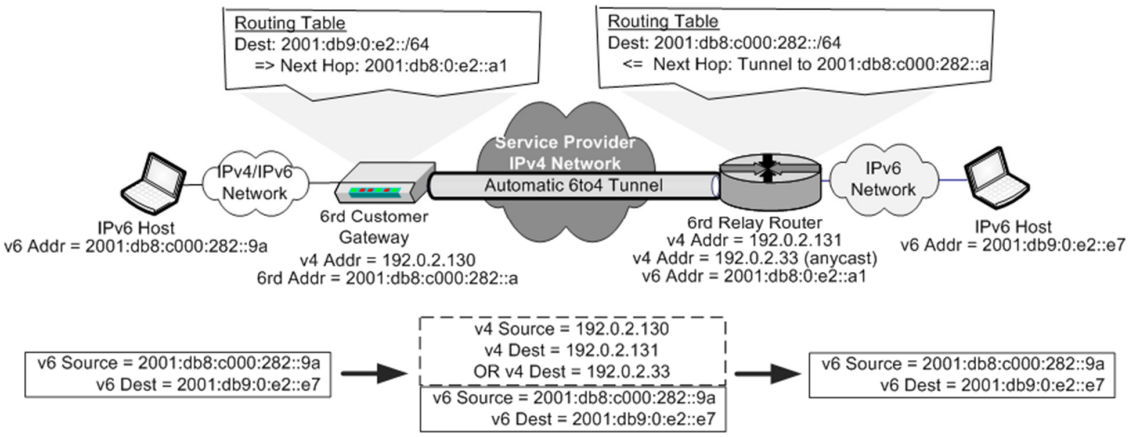
се основава на изграждане на тунели и предаване на IPv6 трафик от CE до IPv6 възел чрез надградена 6to4 технология. 6rd запазва основното предимство на 6to4 – улеснено изграждане на автоматични тунели и избягва част от недостатъците му. Промените се изразяват в следното:

- За разлика от 6to4, при 6rd е дефинирано, че пакетите, постъпващи от глобалния Интернет са само и единствено за клиенти на доставчика и преминават през 6rd шлюза му.
- Доставчикът може да има един или няколко 6rd шлюза. Шлюзовете трябва да са с anycast адрес 192.88.99.1.
- Добавена е поддръжка на клиенти с частни IPv4 адреси. В този случай пакетите към IPv6 с частни IPv4 адреси биват насочени не към IPv4 NAT устройството, а директно към 6rd шлюз.

Префиксът 2002::/16, дефиниран от 6to4 е заменен с IPv6 префикс, който е част от адресното пространство на оператора. Например 6to4 префикса би изглеждал по следния начин 2002:{32-bit IPv4 address}::/48, а 6rd префиксът {32-bit service provider IPv6 prefix}:{32-bit IPv4 address}::/64.

На Фигура 1-28 е демонстриран оператор с IPv6 блок от адреси 2001:db8::/32 и клиент CE с адрес 192.0.2.130. Използваният от оператора префикс за 6rd е 2001:db8:c000:282::/64.

Фигура 1-28 Пример за използването на 6rd



Клиентското устройство (2001:db8:c000:282::9a) в най-лявата част на схемата инициира изграждането на сесия към устройството в най-дясната част на схемата (2001:db9:0:e2::e7). CE устройството играе роля на 6rd клиентски шлюз и предава в изграден тунел пакетите към предварително конфигуриран 6rd Relay router (препредаващ маршрутизатор) с anycast адрес 192.0.2.33. Маршрутизаторът играе ролята на крайна точка на тунела и препредава пакетите към крайния адрес.

1.5.7.2 6over4 (IPv6 тунел върху IPv4 мрежа)

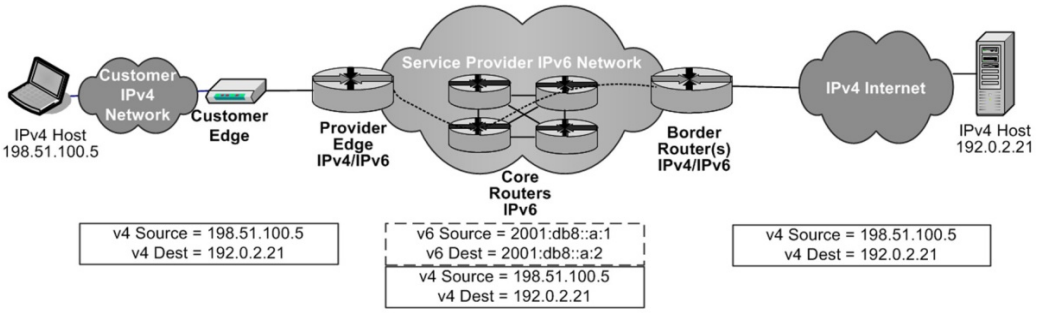
6over4 е механизъм [48] за енкапсулация на IPv6 трафик в IPv4 пакети. Основното изискване на 6over4 е IPv4 средата да поддържа навсякъде "multicast" трафик и поне два възела от нея да поддържат 6over4. По този начин механизмът третира IPv4 мрежата като един единствен домейн с IPv4 multicast възможности. 6over4 енкапсулира IPv6 пакетите в IPv4 "multicast" дейтаграми и препоръчва максимален размер на IP пакета от 1480 байта. В препоръката е специфициран точно определен формат на заглавната част на IP пакетите. Дефиниран е и механизъм за автоматично генериране на IPv6 локални адреси от съществуващите вече IPv4 адреси.

Предимствата са лесна имплементация, а недостатъците са свързани с изискването за мултикаст среда (такава просто няма на много места) и увеличено MTU (Maximum Transmission Unit). В много от случаите, съществува риск пакетите да бъдат фрагментирани или просто отстранени.

1.5.7.3 4over6 (IPv4 тунел върху IPv6 мрежа)

4over6 [49] е механизъм за автоматизирано изграждане на IPv4 тунели върху IPv6 среда, реципрочен на 6over4. Целта е IPv4 абонатите да могат да достигат до IPv4 цели през IPv6 опорна инфраструктура. За разлика от 6PE (разгледан по-долу), при който мрежата на доставчика трябва да бъде MPLS базирана, при 4over6 това не е необходимо. Изисква се обикновена IPv6 маршрутизация.

Фигура 1-29 Пример за използването на 4over6



На Фигура 1-29 Customer Edge обявява мрежа 198.51.100.0/24 на свързания към него Provider Edge. От своя страна PE (Provider Edge) устройството е с двоен IP стек и обявява новия префикс по MP-BGP (Multi-Protocol Border Gateway Protocol) на останалите PE устройства. По същия начин в PE и съответното CE устройството на получателя (не са показани изрично на фигурата) обявяват 192.0.2.0/24.

При пристигане на пакет за 192.0.2.0/24 от CE, PE устройството го енкапсулира в IPv6, така че пакетът да може да бъде маршрутизиран през IPv6 опорната мрежа. От другата страна пакетът се декапсулира и препредава по IPv4 към страната получател (IPv4 Host – 192.0.2.21).

Като предимство на механизъма може да се изтъкне факта, че не е необходимо опорната мрежа да поддържа MPLS, а като недостатък факта, че работи само за една единствена BGP автономна система и че не поддържа “multicast”.

1.5.8 ISATAP (Intra-Site Automatic Tunnel Addressing Protocol)

ISATAP [50] е протокол за автоматично изграждане на тунели между възли с двоен IP стек върху IPv4 мрежова среда. За разлика от 6over4, ISATAP разглежда IPv4 мрежата като NBMA (Non Broadcast Multiple Access) среда за изграждане на IPv6 връзки. Това става чрез вграден механизъм за генериране на локален адрес на IPv6 връзката (link local) на базата на съществуващия IPv4 адрес и чрез ръчно задаване (конфигуриране) на съседните ISATAP устройства.

Основното предимство на ISATAP пред 6over4 е, че IPv4 средата не е необходимо да поддържа трафик от точка до много точки (multicast).

ISATAP се поддържа от последните версии на Windows, Linux и BSD (Berkeley Software Distribution) Unix и е един добър вариант за миграция за всяка една корпоративна организация с развита вътрешна IT инфраструктура.

1.5.9 Teredo

Teredo [51] е протокол, разработен от Microsoft, за автоматизирано изграждане на тунели между IPv4 кандидат за IPv6 свързаност и IPv6 домейн. Протоколът предвижда механизъм за установяване параметрите на тунела и енкапсулация на IPv6 трафика в IPv4 пакети върху UDP протокол. Предимствата на Teredo са:

- Позволява работа в NAT/PAT среда за разлика от механизмите, използващи протокол 41.
- Работи с частни адреси.

Като недостатък може да се изтъкне, че IPv6 MTU се редуцира поради появата на UDP заглавна част.

Teredo е добър вариант за всяка една организация, желаеща да въведе IPv6, независимо от доставчика си на услуги.

1.5.10 6PE

6PE се основава на архитектура, при която PE маршрутизаторите са с двоен IP стек, а MPLS опорната мрежа е IPv4 базирана [52]. Клиентите може да са IPv4 или IPv6 като и в двата случая маршрутизиращата информация се пренася от MP-BGP. Важно е да се отбележи, че 6PE реално поддържа само глобалната маршрутизираща таблица. В равнината за данни се добавят минимум два етикета. Външният е за изграждане на път с комутация на етикети в опорната мрежа, а вътрешният се използва от 6PE механизъма за комутация на IPv6 трафика. 6PE е добър вариант на механизъм за пренос на Интернет IPv6 трафик и маршрутизация върху съществуващата MPLS инфраструктура без да се налага опорната мрежа да поддържа IPv6. 6PE не е подходящ за предоставяне на IPv6 MPLS VPN услуги, поради факта, че не предлага разделение на маршрутизиращата информация в отделни (виртуални) таблици. За целта съществува друг механизъм - 6VPE.

1.5.11 6VPE

6VPE е базиран на MPLS L3 IPv4 VPN и позволява един виртуален маршрутизиращ домейн VRF (Virtual Routing and Forwarding) да поддържа както IPv4, така IPv6 [53]. Подобно на традиционния MPLS L3 VPN, механизмът се основава на MP-BGP за пренос на информацията от контролната равнина и на допълнителни етикети, идентифициращи отделните VRF инстанции в равнината за данни. Механизмът разрешава ситуацията с предоставянето на виртуални частни мрежи, поддържащи IPv6 върху съществуваща MPLS IPv4 опорна инфраструктура.

1.6 Основни изводи, получени в резултат от направения цялостен литературен обзор

В Глава 1 е направен обзор на:

- процеса на еволюция на комуникационния процес от дълбока древност до наши дни;
- приликите и разликите между IPv4 и IPv6;
- механизмите за раздаване на адреси и за преход от IPv4 към IPv6.

В заключение от направения обзор може да се обобщи, че процесът на еволюция на комуникационния процес е прогресирал едновременно с обществото. Той е бил повлиян от множество социално-технически събития и открития и в крайна сметка се е стигнало до една глобална мрежа – Интернет. Интернет от своя страна е разделен на множество по-малки автономни подмрежи, управлявани от различни организации.

Съвременният Интернет е силно зависим от IP и по-точно от текущата му версия 4. IPv6 е протокол, предлагащ редица предимства спрямо IPv4 и е достатъчно зрял за да замени IPv4 в съвременните комуникационни мрежи [7], [8], [21], [54]**Invalid source specified.**.

За целта са дефинирани са множество механизми за преход. Няма механизъм който да работи еднакво добре във всяка една среда и ситуация [4], [8]. Измислянето на нови и нови механизми прави вземането на решение, кой е най-подходящия спрямо даден контекст все по-сложна задача.

Като следствие от това въпреки, че протокола е почти на 20 години, дяла на IPv6 все още е сравнително малък и не се наблюдава масова миграция [3], [4], [7], [5], [55].

Причината за това се корени във факта, че версията на IP протокола е част от системните свойства на всеки един възел и на всяка една връзка във почти всяка съвременна мрежа [56]. Протоколът е интегриран с почти всяка една преносна технология и почти всяка една услуга предоставяна от кой да е доставчик [4], [8] [54].

За да бъде решен проблемът с прехода, първо той трябва да бъде пречупен през призмата на контекста и зависещи от заинтересованите лица функционалните изисквания, качествените характеристики, бизнес и техническите ограничения свързани от всяка една система [57]. Глобалната мрежа е система от системи и това прави дефинирането на изискванията на изключително сложен за решаване проблем [58]. Добре би било да бъде наложен единен подход, съобразен с контекста и изискванията наложени от различните заинтересовани лица. В резултат от подобен подход и анализ да бъде създадена и оценена най-подходящата стратегия за прилагане на механизмите върху мрежата.

Изпълнението на подобни стратегии трудно би могло да се извърши ръчно от един или няколко инженери в голяма IT инфраструктура (над 1000 възела). Съвременните мрежи са хетерогенни по своята същност и съдържат в себе си множество системи и приложения, произведени от различни производители и подчинени на различни контексти. Това прави задачата по приложение на механизмите върху „живата“ мрежа не по-малко тежък от проблема свързан с намирането на най-подходящите механизми.

Вместо традиционния „ръчен“ подход [4] по-добре би било да се използват софтуерни средства и чрез тях автоматизирано и контролирано да се извърши трансформацията на мрежата от текущото IPv4 състояние към желаното IPv6. Литературният обзор не би бил пълен, без анализ и на системите за управление на мрежата и бизнеса, които изглеждат най-подходящите за целта. Това наложи литературния обзор да бъде разширен и в отделна глава да бъдат разгледани и възможностите на тези системи и решения по отношение на прехода от IPv4 към IPv6.

Глава 2: Обзор и анализ на съвременните системи за управление на мрежата и бизнеса

2.1 Въведение

Системите за управление на мрежата и бизнеса са една от най-важните части в мрежата на всеки един съвременен доставчик на телекомуникационни услуги. Преди години акцентът е бил върху самата мрежа и оборудването, което я изгражда. В момента фокусът е изместен в друга посока - как операторът да управлява ефективно своята мрежа и как да извлича максимален доход от съществуващата инфраструктура. Това е възможно само чрез системи за управление на мрежата и бизнес процесите.

Анализът на архитектурата на системите за управление на мрежите и бизнеса от една страна цели да определи дали те биха могли да бъдат основната движеща сила на еволюционния процес, а от друга да определи как самите те еволюират [59].

OSS (Operation Support Systems) са системи за управление на мрежата, мрежовите инфраструктури, подготвяне на конфигурации за предоставяне на услуги, конфигуриране на мрежови елементи и управление на повреди [60]. BSS (Business Support Systems) са системите за управление на бизнеса, които обхващат дейностите по управление на бизнес процесите на оператора. Примери за процеси, протичащи в BSS са приемане на поръчки, генериране на фактури, събиране на плащания, стартиране на маркетингови кампании и др.

Основните сфери на OSS/BSS може да бъдат обобщени в три групи:

- Order Fulfillment – управление на поръчките, предоставяне на услуги и управление на ресурсите [61].
- Service Assurance – управление на повреди, производителност на мрежата, топология, конфигурация, планиране и тестване.
- Billing – системи за прилагане на тарифи, създаване на фактури, осигуряване събираемостта на приходите.

Сферата на приложение на OSS/BSS не се изчерпва само с изброените дейности. В съвременните телекомуникационни мрежи OSS и BSS се разглеждат като две допълващи

се и непрекъснато взаимодействащи подсистеми на една система. OSS системите са продукти, чиято основна цел е да подпомагат и улесняват процеса по експлоатация на дадена мрежа. Те се използват основно от техническите отдели на съответния мрежов оператор и подпомагат дейностите по управление и наблюдение на мрежовото оборудване. Все още широко разпространени са случаите, в които един мрежов оператор има множество и различни OSS системи, които се използват от различните технически отдели. Наличието на множество такива системи е обусловено от множеството различни технологии, използвани в мрежите и множеството различни доставчици на оборудване. Например в мрежата си за достъп всеки един оператор може да има няколко различни доставчици на базови станции. Повечето доставчици имат и свои собствени OSS системи, които трудно се интегрират помежду си. В резултат на това, операторът има множество различни системи, като всяка от тях представя своя поглед върху част от мрежата, което е съществен недостатък.

BSS системите от своя страна се използват от отдели, свързани с продажби на услуги, създаване на промоционални пакети, таксуване, издаване на фактури и взаимодействие и грижа за клиента. Често при един и същ оператор, който предлага различни услуги има и различни такива системи. Например за услугите, свързани с пренос на глас, BSS системата е една, а за услугата Video on Demand (Достъп до видео съдържание при поискване) е друга. Както и при OSS системите и тук се получава различен поглед заради различните видове BSS системи и липсата на интеграция по между им. Такъв подход не предоставя цялостен поглед върху услугите, които използва даден клиент, начина по който ги използва и неговото потребление.

В съвременните телекомуникации OSS и BSS не се разглеждат самостоятелно. Фокусът е върху интегрирани OSS/BSS системи, които предоставят цялостен поглед върху бизнес, мрежа, клиенти и услуги. Тези системи събират данни чрез стандартизирани интерфейси от мрежовото оборудване. Форматът на тези данни често е различен и поради тази причина следва допълнителна обработка, която да уеднакви форматите. В последствие от тези данни се изчисляват единни параметри, които отразяват използваемостта на дадена услуга или качеството, с което тя се предоставя. Възможностите за допълнителна обработка на вече събраните данни са многобройни, както и параметрите за оценка, които могат да бъдат изчислени. Крайната цел на тази

допълнителна обработка е да се предоставя единна информация за мрежата, услугите и клиентите, която може да се използват от различните по вид отдели в комуникационния оператор.

Много важно изискване е тази информация да бъде достъпна в реално или близко до реалното време. За съвременния комуникационен оператор е много ценно да може да идентифицира евентуални проблеми в мрежата или в услугите, преди да е постъпило оплакване от даден клиент, като реагира своевременно. Тази реакция може да включва промяна в конфигурация от страна на техническия екип на комуникационния оператор, обаждане от страна на център за работа с клиенти, предлагане на бонус към закупения пакет услуги, в случай че клиентът не е могъл да го използва по вина на комуникационния оператор и други. Всички тези дейности са част от OSS/BSS процесите, които протичат в един комуникационен оператор.

Поради големия брой производители на OSS/BSS системи, много важно изискване към тях е улесненото им интегриране. За постигане на това изискване съществуват организации, които дефинират модели и препоръки по отношение на тяхната обща функционалност и интерфейси за взаимовръзка. Основно място сред тези организации, заемат ITU и TM Forum (TeleManagement Forum). ITU-T публикува няколко препоръки, свързани с процесите по управление на телекомуникационни мрежи, а TMForum създава модела eTOM и стандарта NGOSS.

2.2 Препоръки на ITU

През 1996 ITU публикува стандарт M.3010, в който е представена концепция за управление на телекомуникационна мрежа TMN (Telecommunication Management Network) [62]. Тази концепция дефинира четири нива на управление:

- функционално;
- физическо;
- информационно;
- логическо.

Логическото ниво на управление се състои от [62]:

- слой за управление на бизнеса (Business management layer - BML);

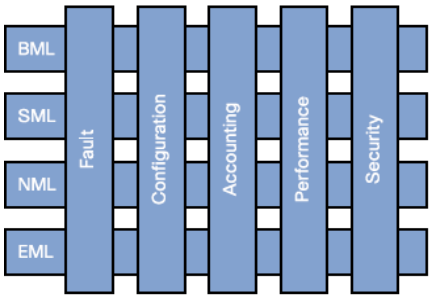
- слой за управление на услугите (Service management layer - SML);
- слой за управление на мрежата (Network management layer - NML);
- слой за управление на елементите (Element management layer - EML).

През 1997 към концепцията за логическото ниво се добавя допълнително разделение на процесите, характерни за комуникационния оператора. Това е описано в препоръка M.3400 [63]. Процесите са групирани по следния начин:

- управление на процеси, свързани с повреди (Fault);
- управление на процеси, свързани с конфигурация (Configuration);
- управление на процеси, свързани с таксуване (Accounting);
- управление на процеси , свързани с оценка на ефективността (Performance);
- управление на процеси, свързани със сигурност (Security).

Структурата на логическото ниво на TMN е представена Фигура 2-1.

Фигура 2-1 TMN - Логическо ниво



TMN е модел за управление на телеком мрежи, който никога не е бил насочен към практическите потребностите на телеком бизнеса. Силата на TMN е била във времето, когато телекомите са били монополисти и не е имало голяма конкуренция на пазара на телекомуникационни услуги. Основния недостатък на TMN е, че не е помислено как бизнес процесите в оператора да бъдат съпоставени с управлението на мрежите [64].

2.3 Стандарт NGOSS

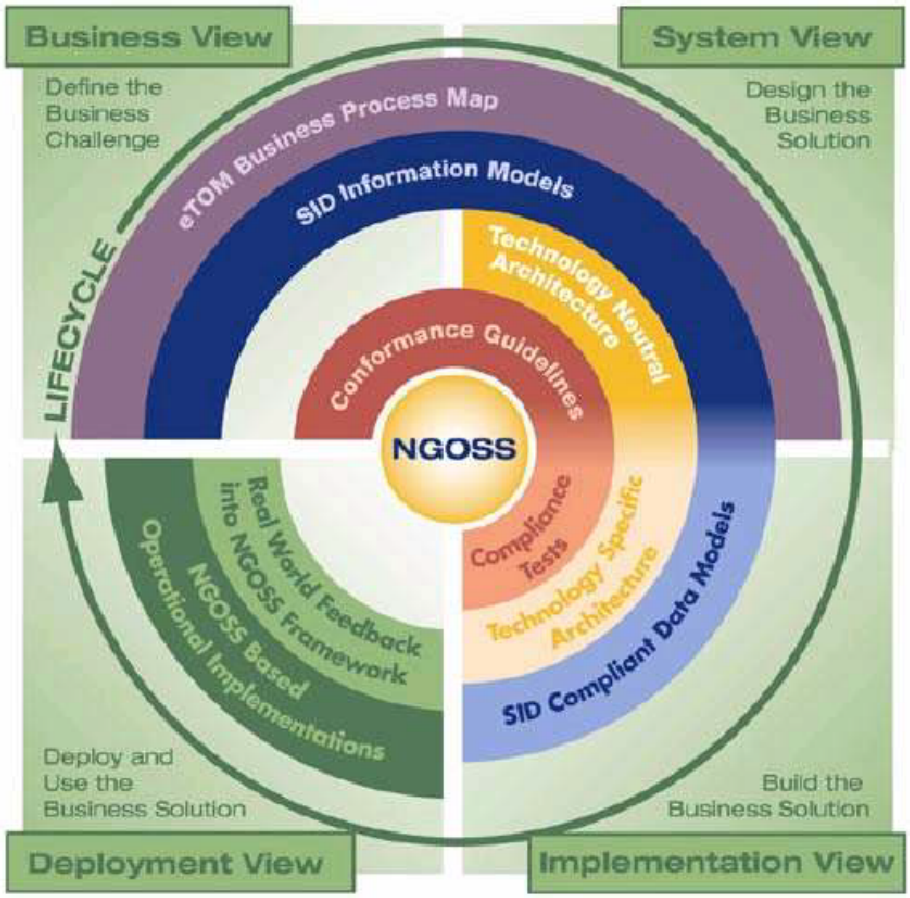
NGOSS (New Generation Operations System and Software) представлява изчерпателна интегрирана структура за разработване, осигуряване и имплементиране на OSS/BSS системи и софтуер.

NGOSS е набор от спецификации и препоръки, които обхващат основни бизнес и технически сфери, сред които са:

- eTOM (Enhanced Telecom Operations Map) – разширен набор от описания на интегрирани бизнес процеси, създадени за клиентски ориентиран пазар с цел управление и анализ на експлоатационния процес;
- модел за споделена информация и данни SID (Shared Information/Data) – този модел съдържа стандартни дефиниции, които определят модела на данните и интерфейсите между отделните приложения в NGOSS;
- създаване на независима от технологиите архитектура (Technology Neutral Architecture) – съдържа основни препоръки и спецификации по отношение на архитектурата;
- дефиниране на критерии за съвместимост и съответствие (Compliance and Conformance Criteria) – съдържа препоръки и тестове, които да осигурят съвместимост на системите, използващи NGOSS;
- създаване на жизнен цикъл и методология (Lifecycle and Methodology) – дефинира процеси и продукти, чрез които интегратори и разработчици да създадат NGOSS приложения, използвайки стандартен подход.

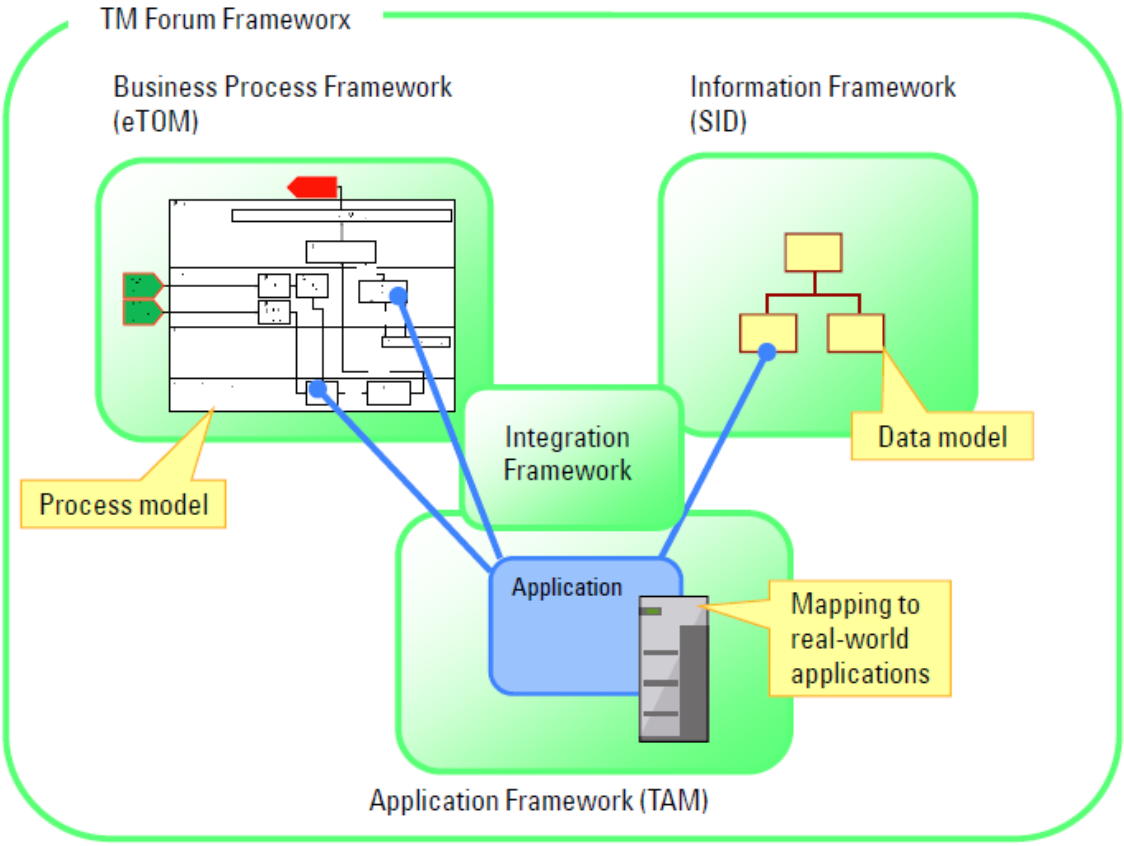
На Фигура 2-2 е представена структурата на NGOSS и различните сфери от няколко гледни точки – от страна на бизнеса, от страна на системите, от страна на разработчиците и от страна на клиентите [65].

Фигура 2-2 NGOSS – Структура



SID предоставя бизнес ориентирана перспектива към модела на данните необходими, за да работи дадена организация. eTOM дефинира процесите, които ще използват данните, а TAM (Telecom Application Map) специфицира приложенията, които ще имплементират тези процеси (Фигура 2-3) .

Фигура 2-3 Интеграция между eTOM, SID и TAM



NGOSS предоставя следните предимства на различните участници в бизнес процеса:

- Доставчици на услуги
    - Улеснява имплементирането на рентабилни OSS/BSS решения чрез стандартни спецификации и интерфейси.
    - Предоставя дългосрочна IT перспектива чрез специфициране на структура на бизнес процесите.
    - Предоставя логическа системна структура, независима от услуги и доставчици на услуги, чрез която IT системите могат да използват бързо развиващи се интегрирани услуги.
    - Намалени експлоатационни разходи чрез по-тясно обвързани бизнес процеси и автоматизация.
- Производители на OSS
    - По-ниска цена за разработка поради използване на стандартна структура.

    - Улеснено взаимодействие с други компоненти чрез използване на стандартни интерфейси.
- Системни интегратори
    - Улеснява имплементирането на нови проекти чрез използване на стандартни компоненти.

Целта на NGOSS е бързо разработване на гъвкави, евтини решения, които да отговарят на бизнес нуждите на доставчиците на услуги. NGOSS може да бъде приложена в организационната структура на доставчика на услуги за следните дейности:

- Изготвяне на нов дизайн на бизнес процесите – eTOM се използва за анализ на съществуващи процеси, откриване на дублиране и пропуски в тях, изготвяне на нови процеси и добавяне на автоматизация.
- Изготвяне на дизайн и специфициране на OSS/BSS решения – NGOSS дефинира детайлен информационен модел, интерфейс и архитектурни спецификации, които може да се използват от доставчиците на услуги да предвиждат и осигуряват бъдещи решения.
- Разработване на приложен софтуер за OSS/BSS, който е съвместим с NGOSS.
- Интеграция на системи – със своя добре дефиниран бизнес и системен език, интерфейси и архитектура, NGOSS дава ясни инструкции при интегриране на системи от различни производители [65].

2.3.1 TMForum eTOM

По същото време, в което ITU-T публикува M.3400, организацията TM Forum създава модел наречен Telecom Operations Map (TOM). В периода 2000 – 2002, този модел е допълнен до Enhanced TOM и публикуван от ITU-T като препоръка M.3050 [65] [66].

Основната разлика между eTOM и TMN е в подхода, използван при управление. При TMN, водещи за управлението са мрежата и мрежовите елементи, докато eTOM е създаден с идеята за управление на всички процеси на доставчика на услуги.

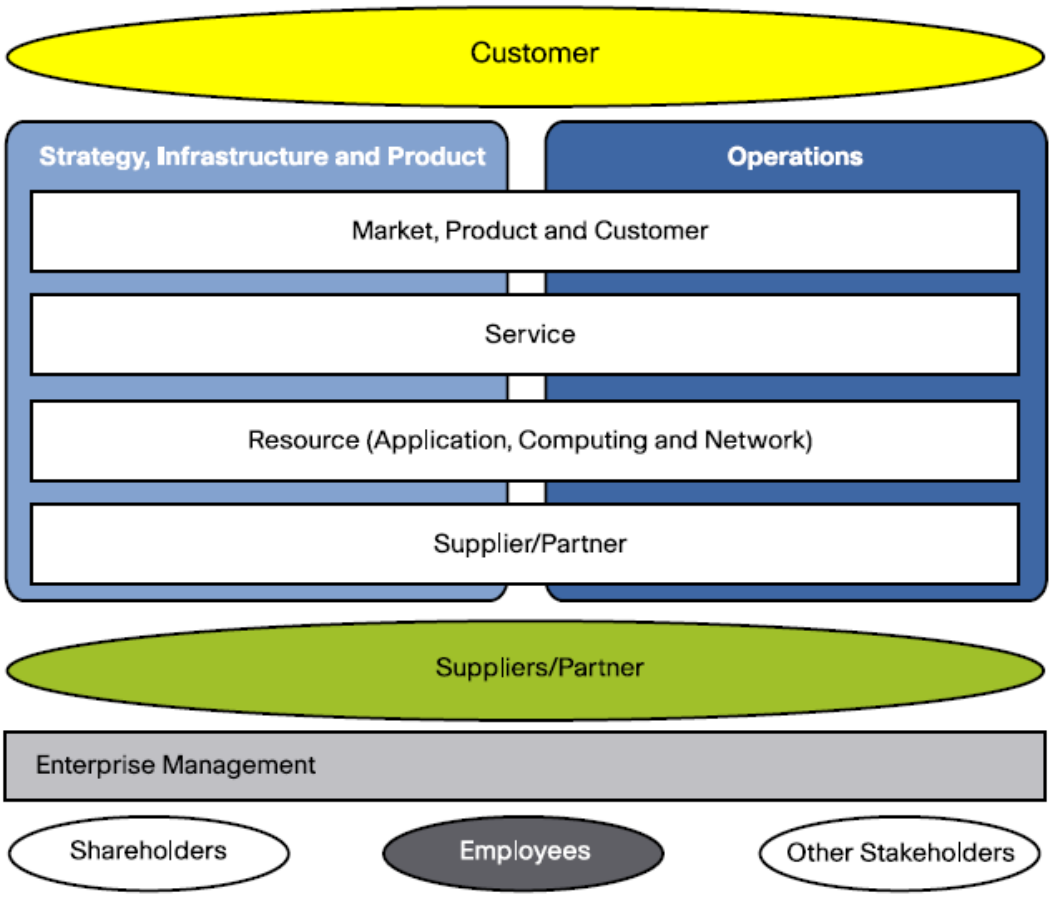
Структурата на бизнес процесите в eTOM описва и анализира различни нива на процеси в зависимост от тяхната значимост и приоритети за бизнес процеса. Тази

структура е дефинирана общо, като целта е да бъде независима от организации, технологии и услуги.

Структурата на eTOM се състои от няколко нива.

Ниво 0 на eTOM (Фигура 2-4) представя функциите на различните отдели в организацията [67].

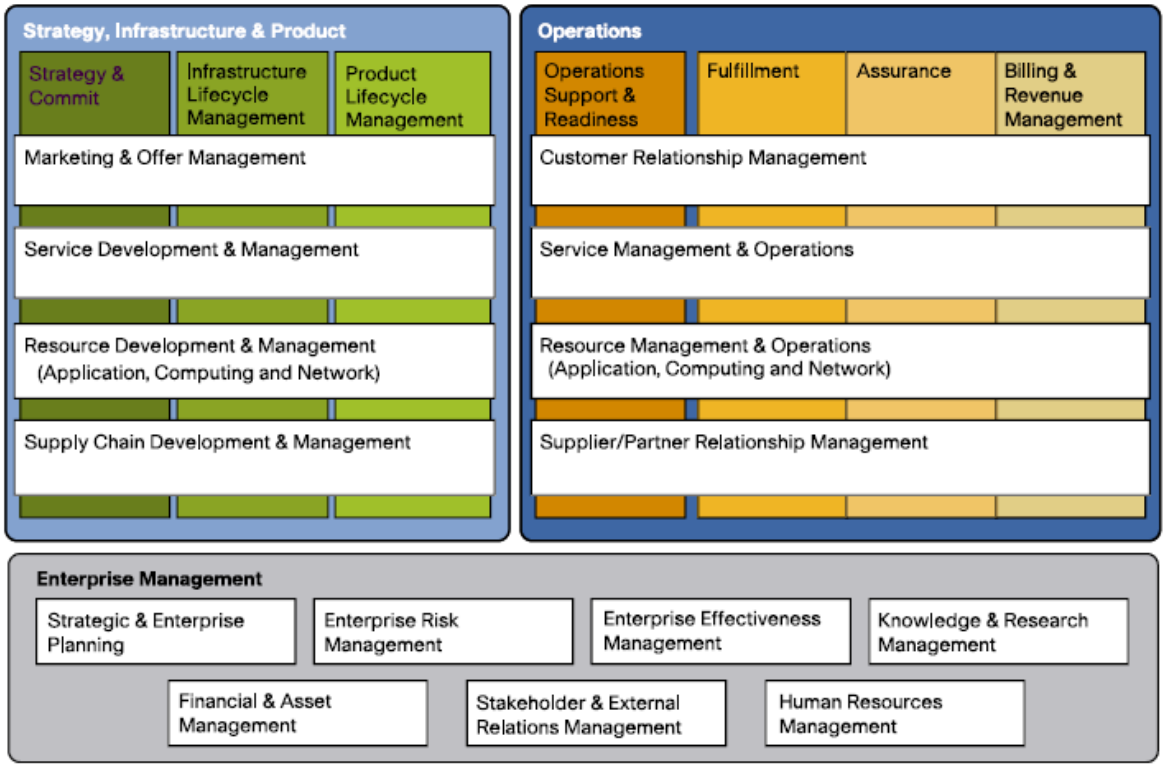
Фигура 2-4 eTOM ниво 0



- Market, Product and Customer (Пазари, продукти и клиенти) – предлага общ поглед върху пазарите и предлаганите продукти.
- Service (Услуги) – това са компонентите на създадените продукти.
- Resource (Application, Computing and Networks) – това са приложенията, използваната изчислителна мощност и мрежите за създаване на услуги.
- Suppliers/ Partners (Доставчици/ партньори) – са всякакъв тип външни компании, от които зависи бизнес процеса на оператора.

Ниво 1 (Фигура 2-5) представя с повече детайли процесите, характерни за комуникационния оператор [67].

Фигура 2-5 eTOM ниво 1



Ниво 1 съдържа седем групи, които обхващат процесите по поддръжка на клиенти и управление на бизнеса [67]:

- Strategy and Commit - Стратегии и изпълнение;
- Infrastructure Lifecycle Management - Управление жизнения цикъл на инфраструктурата;
- Product Lifecycle Management - Управление жизнения цикъл на продуктите;
- Operations support and Readiness - Подпомагане на експлоатацията и осигуряване на готовност за конфигуриране на нови услуги/ клиенти;
- Fulfillment - Създаване и реализиране на продукт/ услуга;
- Assurance - Осигуряване на качество на продукт/ услуга;
- Billing and Revenue Management - Таксуване и управление на приходите.

Всеки един от тези процеси има своя група от процеси, които са насочени към различни дейности по отдели [67]:

- Marketing and Offer Management – Управление на маркетинг и оферти;
- Service Development and Management – Управление и развитие на услуги;
- Resource Development and Management (Application, Computing and Network) – Управление и развитие на ресурси (приложения, изчислителни ресурси, мрежа);
- Supply Chain Development and Management – Управление и развитие на доставките;
- Customer Relationship Management – Управление на взаимодействието с клиенти;
- Service Management and Operations – Управление на услуги и експлоатация;
- Resource Management and Operations – Управление на ресурси и експлоатация;
- Supplier/ Partner Relationship Management – Управление на взаимодействието с доставчици и партньори.

Процесите, свързани с управление на предприятието са следните [67]:

- Strategic and Enterprise Planning – Планиране на стратегии и инициативи;
- Enterprise Risk Management – Управление на риск в предприятието;
- Enterprise Effectiveness Management – Управление на ефективността на предприятието;
- Knowledge and Research Management – Управление на знанията и научно изследователската дейност;
- Financial and Asset Management – Управление на финанси и активи;
- Stakeholder and External Relations Management – Управление на заинтересованите страни и външните взаимоотношения;

- Human Resource Management – Управление на човешките ресурси.

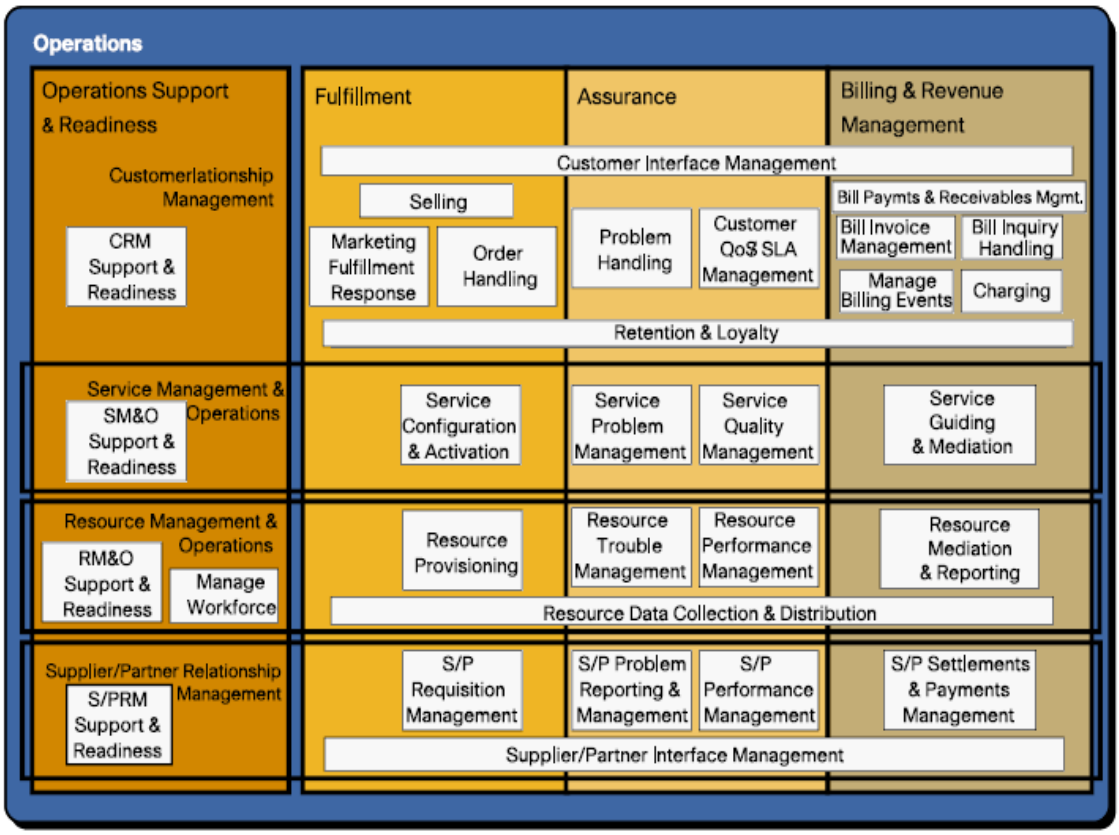
eTOM ниво 2 акцентира върху FAB (Fulfillment, Assurance and Billing) процесите, свързани със създаване и реализиране на услугите, осигуряване на качеството им и тяхното таксуване.

Групата от процеси наречена Operations Support and Readiness (Подпомагане на експлоатацията и осигуряване на готовност за конфигуриране на нови услуги/ клиенти) подпомага пряко тези три основни експлоатационни процеса.

Групата процеси Strategy Infrastructure and Products (Стратегия, инфраструктура и продукти) не са свързани директно с поддържането на клиенти. Те включват изготвяне и прилагане на стратегии, управление на жизнения цикъл на инфраструктурата и продуктите.

Следващите две фигури показват основните процеси на ниво 2. Всеки един от основните процеси е част, както от хоризонтална, така и от вертикална група процеси.

Фигура 2-6 eTOM ниво 2 процеси по експлоатация на мрежата



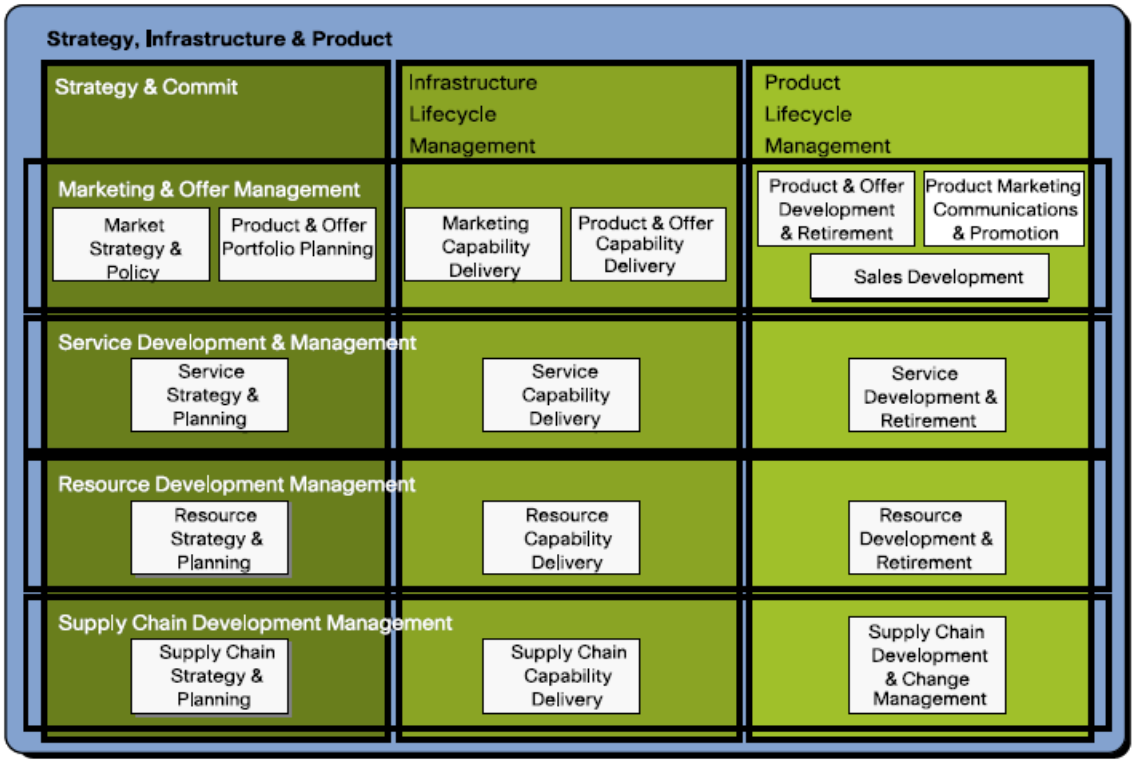
Списък на процесите от eTOM ниво 2 по експлоатация на мрежата [68]:

- CRM Support & Readiness – Поддръжка и готовност на CRM;
- Service Management & Operations Support and Readiness – Поддръжка и готовност при управление на услуги и експлоатация;
- Resource Management & Operations Support & Readiness – Поддръжка и готовност при управление на ресурси и експлоатация;
- Manage Workforce – Управление на работната сила;
- Supplier/ Partner Relationship Management Support & Readiness – Поддръжка и готовност при управление на взаимоотношенията с партньори и доставчици;
- Customer Interface Management – Управление на интерфейса към клиента;
- Selling – Продажби;
- Marketing Fulfillment Response – Отговор към маркетинг при завършено активиране на услугата;
- Order Handling – Управление на поръчките;
- Problem Handling – Разрешаване на проблеми;

- Customer QoSA SLA Management – Управление на договореното с клиента ниво на обслужване и качество на услугите;
- Bill Payments and Receivables Management – Плащане на сметки и управление на вземанията;
- Bill Invoice Management – Управление при издаване на фактури;
- Bill Inquiry Handling – Обработка на запитвания по сметки;
- Charging – Таксуване;
- Retention and Loyalty – Запазване на клиенти и лоялност;
- Service Configuration and Activation – Конфигуриране и активиране на услуга;
- Service Problem Management – Управление на проблеми по обслужване;
- Service Quality Management – Управление качество на обслужване;
- Service Guiding and Mediation – Подпомагане при използване на услуга;
- Resource Provisioning – Предоставяне на ресурс;
- Resource Trouble Management – Управление на затруднения, свързани с ресурси;
- Resource Performance Management – Управление на производителността на ресурсите;
- Resource Mediation and Reporting – Използване на посредничество за ресурси и създаване на отчети;
- Resource Data Collection and Distribution – Събиране и разпределяне на данни за ресурси;
- Supplier/ Partner Requisition Management – Управление на заявки, свързани с партньори и доставчици;
- Supplier/ Partner Problem Reporting Management – Управление на отчети за проблеми, свързани с партньори и доставчици;
- Supplier/ Partner Performance Management – Управление на производителността на партньори и доставчици;
- Supplier/ Partner Settlements and Payments Management – Управление на договореностите и плащанията, свързани с партньори и доставчици;

- Supplier/ Partner Interface Management – Управление на интерфейса към партньори и доставчици.

Фигура 2-7 eTOM ниво 2 процеси по стратегия, инфраструктура и продукт



Списък на процесите от eTOM ниво 2 за стратегия, инфраструктура и продукт [68]:

- Market Strategy and Policy – Маркетинг стратегии и политики;
- Product and Offer Portfolio Planning – Планиране на портфолио от продукти и оферти;
- Service Strategy and Planning – Планиране и стратегии за услуги;
- Resource Strategy and Planning – Планиране и стратегии за ресурси;
- Supply Chain Strategy and Planning – Планиране и стратегии за доставки;
- Marketing Capability Delivery – Създаване на маркетинг възможности;
- Product and Offer Capability Delivery – Създаване на възможности за продукти и услуги;
- Service Capability Delivery – Създаване на условия за доставяне на услуги;
- Resource Capability Delivery – Създаване на условия за доставяне на ресурси;

- Supply Chain Capability Delivery – Създаване на условия за извършване на доставки;
- Product and Offer Development and Retirement – Развитие на продукти и оферти и прекратяване на стари такива;
- Product Marketing Communications and Promotion - Комуникация и промоции във връзка с продуктов маркетинг;
- Sales Development – Развитие на продажбите;
- Service Development and Retirement – Развитие на услуги и прекратяване на стари такива;
- Supply Chain Development and Change Management – Развитие на мрежата от доставчици и управление на промените.

Всеки един от процесите на ниво 2 е изграден от множество подчинени процеси. Това е постигнато чрез анализ на всеки процес и разделяне на функционалността му на съставни процеси. Тази процедура може да бъде приложена и за по-ниските нива. Най – общо нивата, описани в eTOM модела, могат да се опишат по следния начин [65]:

- ниво 0 – бизнес дейности, които разграничават процесите, свързани с клиентите от процесите за управление и стратегии;
- ниво 1 – процеси, които включват бизнес функции и стандартни процеси от край до край;
- ниво 2 – основни процеси, които се комбинират помежду си за предоставяне на услуги и други процеси от край до край;
- ниво 3 – дейности и асоциирани утвърдени модели в бизнес процесите;
- ниво 4 – стъпки и асоциирани детайлни експлоатационни процеси;
- ниво 5 – по-нататъшно разделяне на съставни процеси, когато е необходимо.

eTOM намира широко приложение сред доставчиците на услуги, поради следните факти:

- предоставя стандарти, терминология и класификация за описание на бизнес процесите и съставните им части;

- предоставя възможност за прилагане на единен стандарт по отношение на разработваните бизнес процеси в предприятието;
- обуславя създаването на последователни и качествени процеси от край до край, с възможност за подобряване на цената и производителността и използване на вече готови процеси и системи;
- предоставя възможност за по-лесно интегриране на приложения.

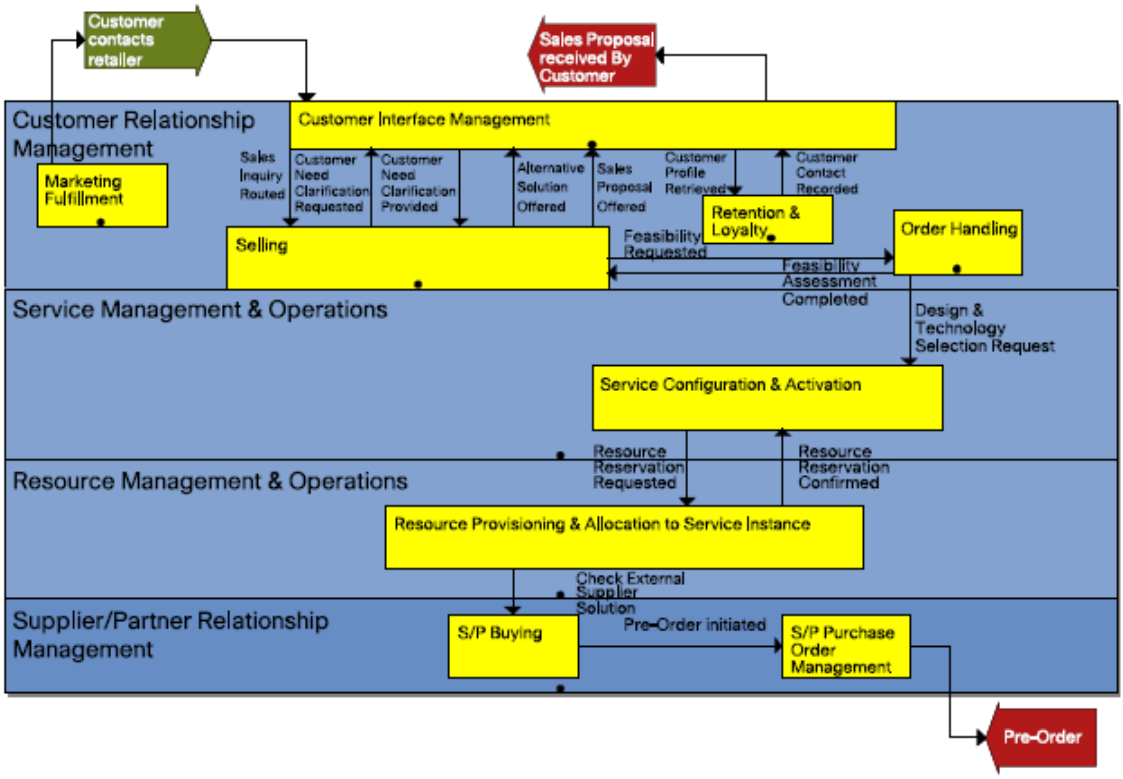
eTOM се фокусира върху процесите, характерни за доставчиците на услуги, връзките между тях, използваните интерфейси, услуги, взаимодействието с клиенти, доставчици, партньори.

eTOM може да се използва за анализ на съществуващи процеси с цел откриване на пропуски, елиминиране на повторения и оптимизиране на процеси. Също така eTOM се използва за създаване на нови процеси, използвайки цялата структура на eTOM, част от нея или допълвайки я. Допълването на eTOM става чрез дефиниране на допълнителни съставни процеси на ниво 3 и ниво 4.

Основни техники за анализ на процесите в една организация са чрез анализ на взаимодействието на процесите и анализ на тяхното протичане.

Пример за протичането на един процес е представен на Фигура 2-8 [65] .

Фигура 2-8 Пример за протичане на eTOM процес



В резултат от работата на отдела по продажби (Marketing Fulfillment), клиентът се свързва с търговец на дребно (Customer Contacts Retailer). Чрез интерфейса за управление на взаимодействието с клиенти (Customer Interface Management) постъпва запитване (Sales Inquiry Routed) към търговския отдел (Selling). Той има нужда от допълнителна информация за нуждите на клиента (Customer Need Clarification Requested). За изясняването им се използва отново интерфейса за управление на взаимодействието с клиенти (Customer Interface Management). След получаване на отговор за нуждите (Customer Need Clarification Provided), търговският отдел изготвя алтернативно предложение (Alternative Solution Offered) и оферта за продажба (Sales Proposal Offered). За да изготви крайното предложение, търговският отдел може да направи запитване към отдела, който обработва поръчките (Order Handling) и да провери до колко е възможно да се изпълни технически търговското предложение (Feasibility Requested/ Feasibility Assessment Completed). След като бъде готова офертата, интерфейсът за управление на взаимодействието с клиенти я изпраща към клиента (Sales Proposal received by Customer). За комуникиране с клиента, Customer Interface Management изпраща и получава

информация от отдела, който се грижи за запазване на клиенти и лоялност (Retention and Loyalty). Тази информация е свързана със запитване за профила на клиента (Customer Profile retrieved) или получаване на контактна информация за връзка с клиента (Customer Contact Recorded).

За да изпълни поръчката, отделът по изпълнение на поръчките (Order Handling) комуникира с отдел за конфигуриране и активиране на услуги (Service Configuration and Activation). Това от своя страна води до резервиране и потвърждаване на необходимите мрежови ресурси (Resource Reservation Requested/ Confirmed). Тази комуникация е между отдела по изпълнение на поръчките и отдела, който предоставя и резервира ресурси за услуги (Resource Provisioning and Allocation to Service Instance). За да предоставя и резервира ресурси, отделът може да комуникира с външен доставчик/ партньор за закупуване на допълнителни ресурси чрез Supplier/ Partner Buying. В случай, че има нужда от допълнителни ресурси, те се поръчват чрез Supplier/ Partner Purchase Order Management.

2.3.1.1 Fulfillment

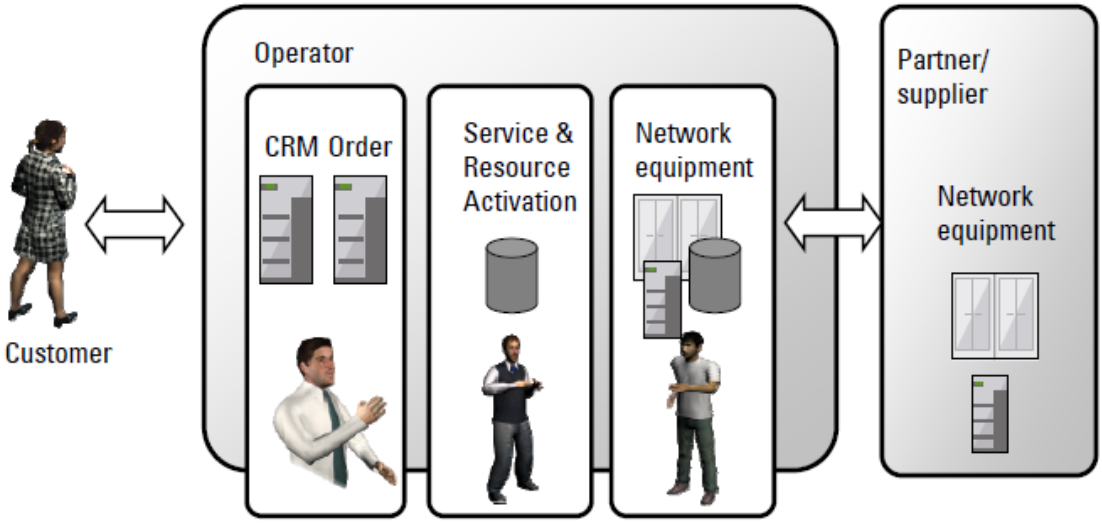
Автоматизацията на процеса от поръчка до услуга започва със запитване от страна на клиента и завършва с доставката на работещ, функциониращ продукт. Правилното протичане на този процес е една от най-важните задачи на всяка една компания с бизнес, свързан с доставката на услуги и реализацията на продукти. Автоматизацията на процеса позволява той да бъде реализиран за по-кратки срокове и с по-малко грешки. По кратките срокове водят до по-скорошно начало на цикъла по таксуването на клиента, а по-малкото грешки до по-високо качество на доставените продукти и услуги. В крайна сметка автоматизацията на този процес е жизнено важна задача за всеки един съвременен мрежови оператор.

Написаното по-горе звучи добре и предполага безпроблемно протичане на процеса по автоматизация на реализацията на продукти и услуги. Реалността е точно обратната. Често срещана практика е забавянето на доставка или реализация на продукт, различен от този, който клиентът предварително си е поръчал. Причините за подобни проблеми са в сложността и многообразието на съвременните мрежови инфраструктури и предлаганите върху тях продукти и услуги. Процесът се разделя на множество съставни процеси и зависи от значително количество предварително събрани данни. Данните трябва да

отразяват достатъчно точно текущото състояние на мрежовата инфраструктура и да моделират в пълна степен желаната от клиента услуга.

На Фигура 2-9 е представено обобщение на процеса по реализация на услугите. Интерфейсът с клиента се управлява от системи за управление на взаимоотношенията с клиента CRM (Customer Relationship Management). CRM системите подават необходимата информация на система Service Configuration and Activation, която на базата на входящата информация и текущото състояние на мрежата създава услугата за съответния клиент. Веднъж реализирала дадена услуга, системата за управление връща статус „Изпълнено“ на системата за управление на взаимоотношенията, а тя от своя страна уведомява клиента, че услугата му е реализирана и започва процесът по таксуване.

Фигура 2-9 Процес за реализация на услуга на даден клиент



Процесът по изпълнение на поръчките се състои от управление на поръчките, предоставяне на услуга, управление на ресурсите, необходими за реализиране на услугата.

Управлението на поръчките може да се раздели на няколко етапа:

- постъпване на поръчка и валидиране (проверка на параметрите) – на този етап се определя абонаментния пакет или план, адресът на който ще се използва услугата, клиентска сметка, контакти с клиента и договори. Процесът по проверка на параметрите проверява въведените данни. Това може да стане в момента на въвеждане или след като всички данни са въведени.

- разделяне на поръчката на една или няколко заявки в зависимост от поръчаната услуга и количеството.

Самият процес по управление на поръчките може да се окаже доста сложен, поради съществуването на различни системи за различните услуги. Това от своя страна забавя и оскъпява въвеждането в експлоатация на нови поръчки, поради необходимостта от поддържане на различни системи и приложения. За улесняване и ускоряване на процеса по управление на поръчките се използват системи, които поддържат каталог с продукти. Те се състоят от правила за класифициране на услугите според клиентски профил, канали за поръчка, местоположение, зависимости между различни продукти, наличие на услугата или други бизнес условия.

Процесът по предоставяне на услуга включва дейностите по създаване и настройка на конкретната услуга след приемане на поръчката. За тази цел се определят необходимите мрежови ресурси, създават се конфигурации, окабеляване, заделят се ресурси в преносната мрежа. Тази дейност се извършва ръчно чрез графичен интерфейс от оператор или автоматизирано. След като услугата вече е конфигурирана, остава тя да бъде активирана. Активирането може да включва конфигуриране на устройство на място или физическо свързване на линия при клиента, което се извършва от технически персонал на място. Повечето от дейностите по активиране на услуги се извършват автоматично – например конфигуриране на ATM или канални комутатори, DSLAM, кабелни модеми и други. Автоматичните команди се изпращат от системите за активиране към мрежовите елементи, системите за тяхно управление или системите за управление на мрежата. Системите за активиране на услуги си взаимодействат с множество други системи и поради тази причина разполагат с различни адаптери за комуникация. Също така, тези системи поддържат процедури по деактивиране на услуга в случай, че възникне грешка в някоя от стъпките по активиране. Системите за активиране комуникират със системите, които управляват наличните мрежови ресурси (Inventory management). Това взаимодействие гарантира, че необходимите мрежови ресурси за активиране на услуга са налични и след успешно активиране, отразява настъпилите промени в използвани и свободни ресурси.

Системите за управление на мрежовите ресурси поддържат информация за свободни и използвани капацитети, местоположение на ресурсите, техния статус, IP адреси, номера на канали, портове и т.н. Тези системи спомагат за автоматизиране на процесите по дизайн и планиране на мрежата. Съществуват и приложения, които автоматично разкриват използваните и свободни ресурси и сравняват резултата със съхранените данни в системите за управление на ресурси [61].

2.3.1.2 Service Assurance

Системите за Service Assurance имат за цел да осигурят оптимално използване на услуга от клиент, запазване на съществуващите клиенти, добавяне на нови и непрекъснато следене на предоставяното качество, така че да не се налага плащане на неустойки към клиент, в случай на понижено качество. Предоставяното качество на обслужване се дефинира под формата на договор, наречен SLA (Service Level Agreement).

Основните системи, които се използват в Service Assurance са [61]:

- Fault and Trouble Management – управление на повредите. Това са системи, които са създадени да откриват, изолират и коригират повреди в мрежата. Те наблюдават и получават аларми от мрежовите елементи. Пример за такава аларма може да е прекъснат кабел или повреден порт. Повредите обикновено оказват влияние не само върху пряко свързаните с тях мрежови елементи, но и върху други. Също така те мога да повлияят и на различни по вид услуги. Системите за управление на повредите са самостоятелни или вградени в системите за управление на мрежовите елементи. По–старите системи са изпращали само известяване за настъпилата повреда, докато по-новите предоставят детайлна информация за повредата. Системите за управление на повредите получават аларми по SNMP, CMIP или чрез специфични адаптери, в зависимост от различните системи за управление на елементите. След постъпване на алармата за повреда, тя се разпределя за отстраняване според зададения приоритет. Системите за управление на повредите позволяват да се променя първоначално определения приоритет за повредата и тя да бъде ескалирана в случай на нужда. Тези системи може да се конфигурират да изпращат и различни известия по E-mail, SMS, пейджинг или др. Системите

за управление на повредите имат графичен интерфейс, който може да се показва на големи екрани в NOC (Network Operations Center). През този интерфейс операторът в центъра може да се свърже към съответната система за управление на елементите и да извърши тестове или диагностика.

- Performance Management – управление на производителността на мрежата. Тези системи събират различни по вид статистики и ги запазват в база данни. Пример за такива статистики са наличие на услуга, скорост на предаване на данни, закъснение, време за отговор от страна на мрежата, загуби и други. Данните могат да се събират пасивно или активно. При пасивното събиране на данни се използват различни по вид проби или се получават данни по SNMP/ FTP от различни EMS системи (Element Management Systems). При активно събиране на данни се извършват целенасочени тестове и резултатите от тях се съхраняват за анализ. След като бъдат събрани данните, независимо дали пасивно или активно, от тях се изчислят параметри за оценка на производителността и качеството на мрежата и услугите. За тези параметри се определят нормални стойности и по отклонението от тях се прави анализ на качеството. Задават се гранични стойности и при преминаването им се генерират аларми. Аларми може да се генерират и когато уговорено качество за обслужване на абоната не отговаря на действителното. Анализът на качеството може да бъде направен и предварително, като се използват приложения, които да покажат как ще се отрази на мрежата увеличеното натоварване и да подпомогнат процеса по планиране на разширение.
- Topology and Configuration Management – управление на топологията и конфигурациите. В миналото мрежите са били по статични и съответно контрола върху промените е бил по-лесен. В съвременните динамични мрежи се отделя много внимание на процесите по управление на топологии и конфигурации, защото липсата на актуална информация може да доведе до дълги прекъсвания в обслужването. Много мрежови оператори използват приложения за разкриване на текущата топология или конфигурация. Тези приложения може да се стартират веднъж или няколко пъти на ден. Данните за топологията или конфигурацията на устройствата се извличат по SNMP от

поддържаните MIBs (Management Information Base). Наличието на вторични пътища, виртуални частни мрежи и MPLS например усложняват процеса по разкриване на топологията. Процесът по разкриване на топология и конфигурация може да се улесни като се извърши върху част от мрежата, а не върху цялата. Процесът по управление на конфигурацията е особено важен, защото промените може да окажат влияние върху цялата мрежа. Конфигурациите се запазват в база данни или LDAP (Lightweight Directory Access Protocol) сървър. Това позволява възстановяване на стара конфигурация в случай на грешка при имплементиране на промените.

- Planning and testing – планиране и тестване. Тези дейностите са взаимно свързани, защото на база резултатите от проведените тестове се прави планиране на необходимите нови ресурси. Тестването може да се раздели на три основни групи – тестване на съществуваща мрежа или промяна в мрежата, тестване при интегриране на нова услуга за абонат, тестване на услуга от край до край. В процеса на тестване се генерира изкуствен трафик за глас, данни, FTP, HTTP, E-mail услуги. При интегрирането на нова услуга за абонат обикновено се тества, как мрежата е настроена за тази нова услуга – маршрутизиране, заделени преносни ресурси и т.н. След това услугата се тества от край до край.

2.3.1.3 Billing

Системите за таксуване включват процеси по генериране и обработка на информация за таксуване, събиране на приходи за използвани мрежови ресурси или достъп до услуга. За получаване на данни за използваните ресурси се използват междинни системи (Mediation systems). Те събират информация от мрежовите елементи и я предават на процес, който прилага съответните тарифи и генерира съответната такса (Rating). Междинните системи получават записи за повиквания или сесии от данни, които включват информация за начало и край на сесията, участващи абонати, продължителност на сесията. Тези записи може да се изпратят и към други системи, например към отдела, който следи за измами.

Процесите, които генерират такси за използваните услуги са сложни, защото трябва да вземат предвид не само продължителността на сесията, но и съответния тарифен план, който абонатът използва, къде и кога е използвал услугата, има ли право на отстъпки.

Системите за таксуване получават обобщени данни с вече приложените тарифи и създават фактури. Във фактурите присъства и информация за текущ баланс на клиента, еднократни инсталационни такси, такса за наем на линии (ако има такава), активни промоции, отстъпки и кредитни лимити. Комбинирането на цялата тази информация в един документ е един от най-сложните процеси, които протичат във всеки оператор. Съществена функционалност, която трябва да притежават тези системи е възможността в една фактура да се съдържа информация за различни услуги – глас, данни, видео.

Освен изготвяне на фактури към крайни клиенти, това се прави и към оператори или доставчици, с които съответния оператор има партньорство. Преди да може да се генерират такива фактури, много важно е работата на системите от двете страни да се съгласува според договора за взаимна свързаност.

В процеса по таксуване попадат и дейностите, свързани с осигуряване на събираемостта на приходите и откриване на злоупотреби – Revenue Assurance and Fraud Management.

Злоупотребите може да се разделят на две основни групи – злоупотреба с неоторизиран достъп или злоупотреба с вече съществуващ абонамент. За откриване на злоупотреби с неоторизиран достъп, системите правят анализ на използваемостта на услугата по продължителност или дестинации. Целта е да се намери нетипичен трафик, който се различава от досегашното поведение и той да се анализира. В случай, че има съмнения за неоторизиран достъп, системата може да пристъпи към блокиране на някоя от услугите до изясняване на случая. Това се прави с цел минимизиране на финансовите загуби.

При злоупотребата с абонамент, клиентът предоставя неверни данни като име, адрес за използване на абонамента или данни за кредитна карта, с която ще се извършват плащанията. За да се открие подобна измама, трябва да се направи сравнение между предоставените данни от предишни поръчки, ако има такива, и тези от последната поръчка. Друг подход, който използват системите за откриване на измами е сравнение на

клиентския профил на заподозрян нов абонат с типичния профил на нов абонат – какви услуги използва, с каква продължителност.

Друг съществен процес част от Revenue Assurance е процесът Churn Management. Той е изключително важен за услугите, които се предлагат на абонаментна база. Системите, които отговарят за Churn Management съдържат средства, чрез които може да се анализира автоматизирано поведението на абоната и да се предскаже кога той потенциално може да прекрати абонамента си.

Интегрираните системи Shared Information/ Data за OSS/BSS улесняват ежедневната работа на телекомуникационните оператори и доставчици на услуги. Съществен проблем е, че често поради своята сложност тези системи не могат да бъдат интегрирани напълно. Това води до необходимост от използване на оператори и извършване на някои от дейностите ръчно [61].

2.3.2 SID (Shared Information/Data) – Общ информационен модел

SID предоставя интегриран информационен модел за приложенията от NGOSS, съдържащ цялата необходима информация за нормалното функциониране на OSS/BSS в даден мрежови оператор.

SID е фокусиран върху логически елементи наречени “business entities” и дефинира техните характеристики чрез съвкупности от атрибути. “Business entity” е обект, който отговаря на определен бизнес интерес. В телеком домейна това са абонатите, продуктите, услугите, мрежата и изграждащите я устройства. Моделът е базиран на съвкупности от логически единици и взаимоотношения помежду им. Всяка една логическа единица се състои от определено количество данни описани като атрибути и методи за манипулация на данните.

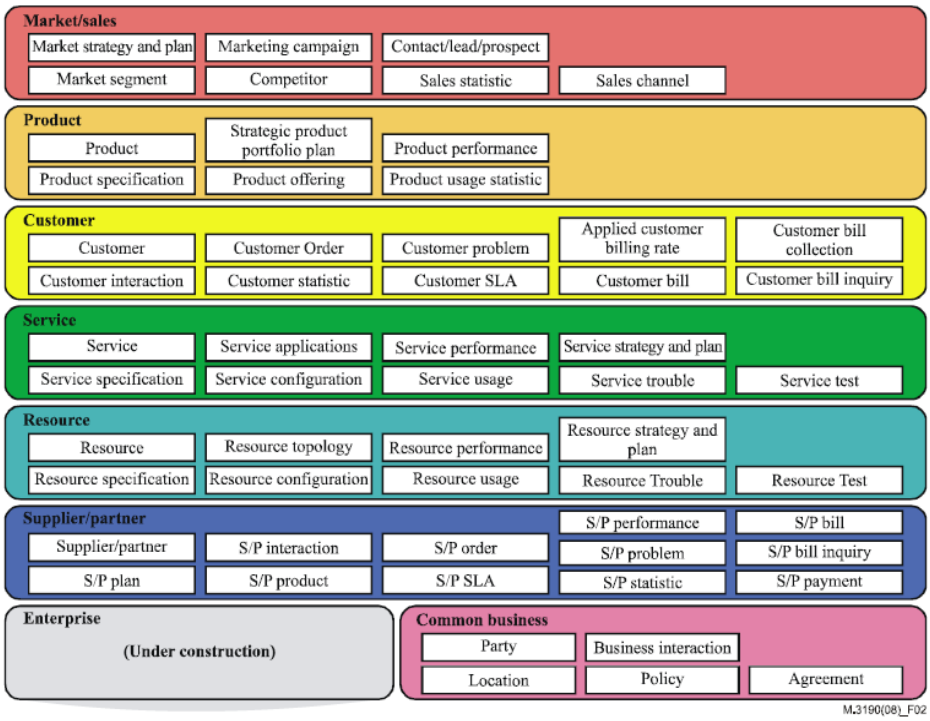
За да бъде интегрирана информацията от всички модели в индустрията в един единствен общ модел, SID е стандартизиран от ITU-T в препоръка M.3190 [69].

Според M.3190 SID модела дава следните предимства пред специфичните подходи, характерни за повечето производители и оператори:

- позволява улеснено управление на информацията чрез обща терминология и избягване на ненужните разлики;

- уеднаквява информационния модел в организацията и между организациите;
- изгражда мост между бизнеса и технологичните отдели в една организация като предоставя дефиниции, които са разбираеми както за бизнеса така и за разработчиците.

Фигура 2-10 SID информационен модел



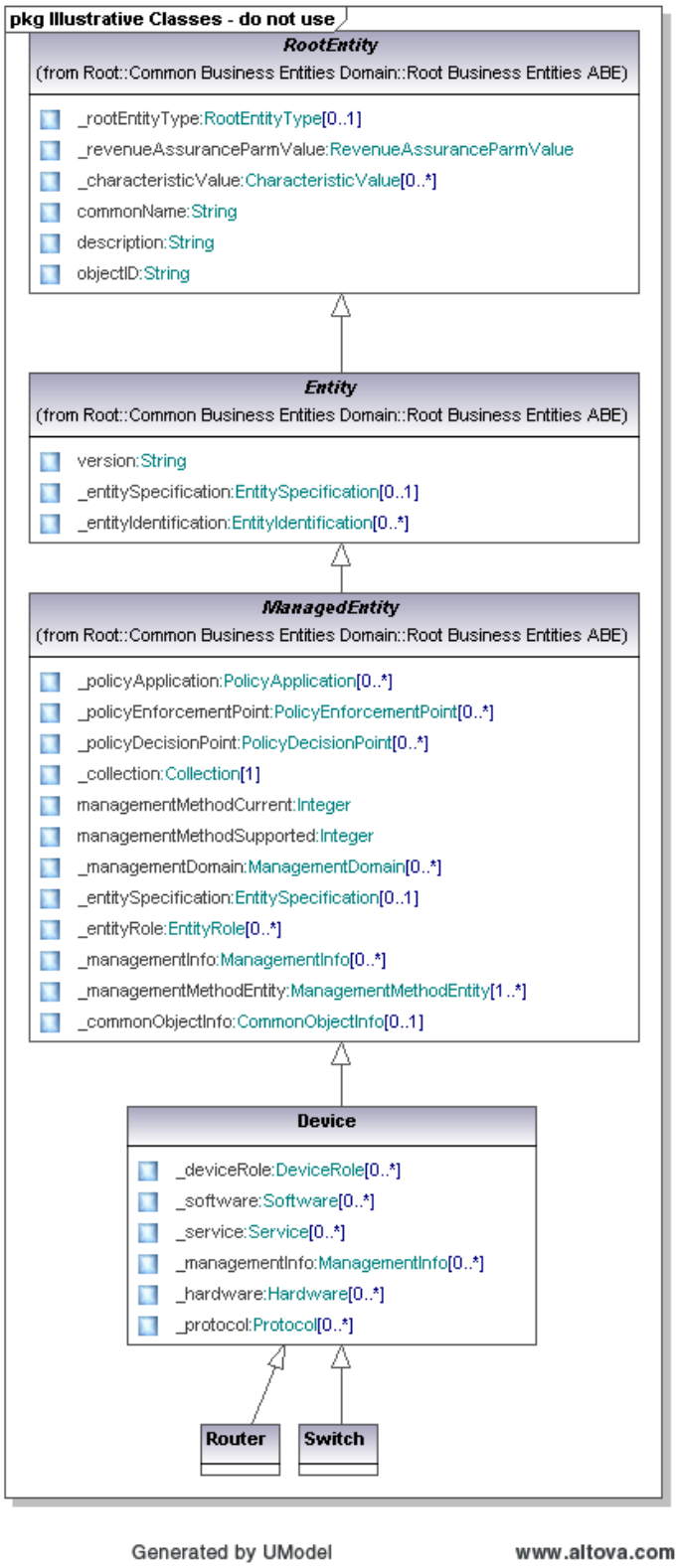
Моделът на данните в SID (Фигура 2-10) е разделен на няколко слоя, които разделят споделената информация в 8 домейна. Всеки един домейн отговаря на област от eTOM (Level 1). Във всеки един от домейните има силна връзка между бизнес логическите единици, които го изграждат. Между домейните има „loose coupling”. Тази подредба позволява сегментацията на бизнес проблема по управлението на мрежата в малки области и позволява всеки да насочи ресурсите си към областта, в която има интерес. С други думи, ако е необходимо да бъде автоматизиран даден бизнес процес, трябва да бъде идентифицирана само информацията, която е необходима на процеса.

Всеки един от осемте домейна се състои от логическите единици наречени ABE (Aggregate Business Entities). Всяко едно ABE може да съдържа по-малки ABE, имащи отношение към по-малки зони от домейна.

Моделът на данни в SID се използва в три големи групи стандарти, специфициращи интерфейси за достъп до модела на данните – OSS/J (OSS through Java), MTOSI (Multi Technology Operation System Interface) и 3GPP (3rd Generation Partnership Project).

Малка извадка от модела, представяща йерархията от обекти от RootEntity до Device (Устройство), е демонстрирана на Фигура 2-11.

Фигура 2-11 SID UML клас диаграма (RootEntity – Router,Switch Devices)


pkg Illustrative Classes - do not use
RootEntity
(from Root::Common Business Entities Domain::Root Business Entities ABE)
_rootEntityType:RootEntityType[0..1]
_revenueAssurancePramValue:RevenueAssurancePramValue
_characteristicValue:CharacteristicValue[0..*]
commonName:String
description:String
objectID:String
Entity
(from Root::Common Business Entities Domain::Root Business Entities ABE)
version:String
_entitySpecification:EntitySpecification[0..1]
_entityIdentification:EntityIdentification[0..*]
ManagedEntity
(from Root::Common Business Entities Domain::Root Business Entities ABE)
_policyApplication:PolicyApplication[0..*]
_policyEnforcementPoint:PolicyEnforcementPoint[0..*]
_policyDecisionPoint:PolicyDecisionPoint[0..*]
_collection:Collection[1]
managementMethodCurrent:Integer
managementMethodSupported:Integer
_managementDomain:ManagementDomain[0..*]
_entitySpecification:EntitySpecification[0..1]
_entityRole:EntityRole[0..*]
_managementInfo:ManagementInfo[0..*]
_managementMethodEntity:ManagementMethodEntity[1..*]
_commonObjectInfo:CommonObjectInfo[0..1]
Device
_deviceRole:DeviceRole[0..*]
_software:Software[0..*]
_service:Service[0..*]
_managementInfo:ManagementInfo[0..*]
_hardware:Hardware[0..*]
_protocol:Protocol[0..*]
Router
Switch
Generated by UModel
www.altova.com

2.3.3 Структура Приложения

Дефинираните по-горе процеси за частта „Експлоатация“ от eTOM трябва да бъдат реализирани в конкретни софтуерни продукти. Структура „Приложения“ дефинира именно съответствието между конкретните приложения и различните видове процеси. Приложенията може условно да се разделят на такива съсредоточени върху процесите, свързани с управлението на клиенти, услуги и процеси [70].

2.3.3.1 Управление на клиенти

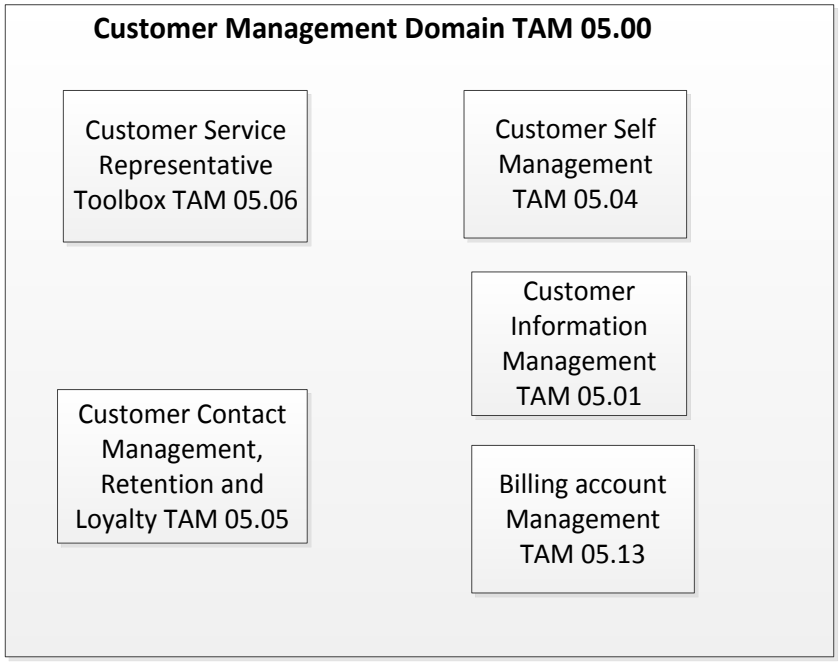
На Фигура 2-12 са представени основните групи приложения, свързани с процесите по управление на клиенти [70].

- Приложения за управление на потребителските услуги (Customer Service Representative Toolbox) – включва приложения за реализация, таксуване и наблюдение на параметрите на реализираните услуги. Приложенията, част от тази група, реализират процесите по обработката на постъпилите поръчки, обработват възникналите изключения, изготвят отчети и фактури и анализират състоянието на определени поръчки или вече работещи услуги.
- Клиентски интерфейс за самоуправление (Customer Self-Management) – в най-общия случай представлява автоматизиран уеб портал, чрез който клиентите могат да се абонират за нови услуги, да следят и влияят на параметрите на вече съществуващите такива. Важно е да се отбележи, че порталите за самоуправление са директно свързани и с таксуването на услугите. Чрез тях клиентът може да заплаща получените от него услуги, а също така и да получи информация за таксуването за даден минал период от време.
- Управление на взаимоотношенията с клиента (Customer Contact Management, Retention and Loyalty). Приложенията от тази група реализират процесите по управление на информацията за взаимоотношенията между клиент и оператор и позволяват изпълнението на стратегии за задържане и поддържане на удовлетворението на клиентите от използваните продукти. Приложенията от тази група се използват основно от служителите, ангажирани с грижата за клиента. Тези приложения също така записват и различните взаимодействия и комуникации между клиент и служителите на доставчика на услуги. По-интелигентните системи

от тази група могат да насочат вниманието на операторите към клиенти, които потенциално имат желание да се прехвърлят към друг оператор и да предложат стратегии за задържане на тези клиенти.

- Управление на информация за клиенти (Customer Information Management) – това приложение е в основата на всяка система за управление на взаимоотношения с клиенти. То позволява създаване, актуализиране, търсене и разглеждане на клиентска информация.
- Приложение за управление на партидите и информацията, свързана с процесите по таксуване на клиентите (Billing Account Management).

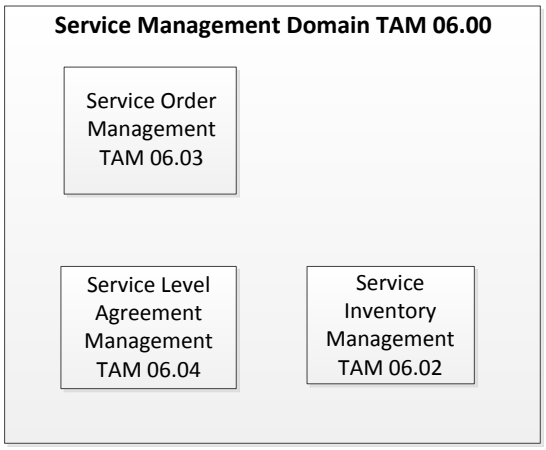
Фигура 2-12 Структура приложения в частта управление на клиенти



2.3.3.2 Управление на Услуги

Втората голяма група приложения е свързана с реализацията на процесите по управление на услуги. Основните приложения, част от тази група, са управление на процеса от поръчка до услуга, управление на информацията, свързана с предоставянето и функционирането на услугите и наблюдение на параметрите по предоставяне на качеството на услугите (Фигура 2-13) [70].

Фигура 2-13 Приложения свързани с управлението на услуги

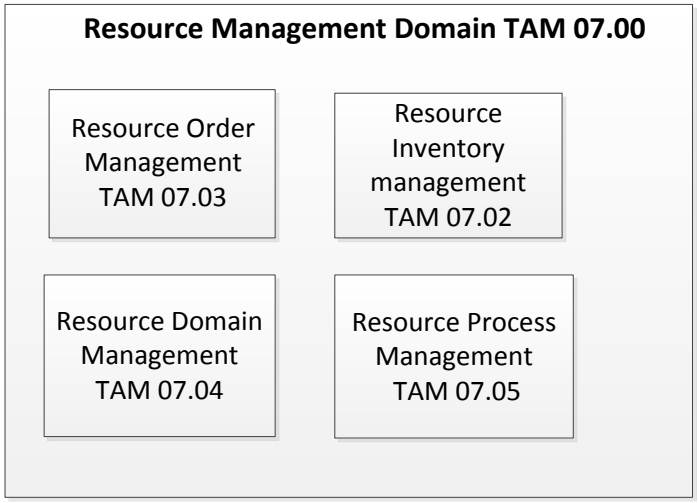


- Управление на база данни с услуги (Service Inventory Management) – използва се за съхранение на детайли за услугите, като например кой ресурс трябва да се използва, за да се активира конкретна услуга. Базата данни съхранява и информация свързана с:
    - връзката между спецификация на продукт и спецификация на услуга и връзката между отделни продукти и услуги;
    - връзката между дадена услуга и компонентите, използвани за нейната реализация.
- Управление на процеса от поръчка до услуга (Service Order Management) – обхваща цялостния процес, свързан с поръчка на услуга и включва:
    - събиране на данни за услугата;
    - проверка за валидност на поръчката;
    - обновяване на статуса на поръчката;
    - управление на прехода от поръчка към конкретна техническа услуга;
    - управление на процеса на проектиране на услугата;
    - управление на процеса по конфигурация и активация на услугата.

2.3.3.3 Управление на ресурси

Услугите условно казано "живеят" в мрежовата инфраструктура на дадения оператор. Всяка една услуга бива реализирана върху един или повече от един мрежови ресурс. Управлението на ресурсите, свързани с процеса по реализация на услугите е представено на Фигура 2-14 [70].

Фигура 2-14 Управление на ресурси



- Приложения за проследяване и наблюдение на ресурси, промяна в конфигурацията на ресурсите, планиране при използване на ресурси, изготвяне на графици, прогнози, съчетаване умения на различен персонал и не на последно място приложенията, свързани с процесите по управление на риска (Resource Order Management).
- Приложения за управление на базата данни с ресурси (Resource Inventory Management) – използват се за управление на информацията за всички използвани ресурси и продукти. Тази подгрупа приложения е свързана с множество системи за управление на състоянието на мрежата и изграждащите я ресурси. Базата данни може да се комбинира с информацията от базата данни за отделните услуги и да се използва за идентифицирането на свободни мрежови ресурси. Пример за подобно взаимодействие е заделянето на свободен мрежови интерфейс за нов клиент. За да се реализира подобна операция, трябва да се установи кои интерфейси са заети.
- Приложение за управление на поръчките за ресурси (Resource Process Management) – това приложение включва генериране на поръчка за ресурс, разделянето на поръчка на компоненти за ресурси или услуги, проследяване на поръчка за ресурс, управление на тестването и активирането.
- Управление на ресурси (Resource Domain Management) – предоставя услуги на всички други приложения свързани с управлението на ресурсите на мрежата. Ресурсите могат да бъдат физически - всички видове мрежови устройства или

логически – адресни пространства, идентификатори, телефонни номера, адреси на страници и др.

2.4 Стандарти, специфициращи как да бъде реализиран NGOSS

OSS/ BSS общността, както всяка друга такава, не е единна по отношение на това, как от стандартизираните модели да се стигне до конкретната реализация на дадени интегрирани продукти. Самата общност е разнородна съвкупност от множество заинтересовани от NGOSS, включително производители на софтуер, оператори и системни интегратори. Всяко едно от тези заинтересовани лица има свой собствен интерес в това, как ще бъде реализиран NGOSS. Например производителите на софтуер вече имат такъв и интересът им е свързан със стандарти максимално близки до техния софтуер. Операторите имат определени типове мрежа и определени софтуерни продукти. Техният интерес е свързан с развитие на стандарт, подходящ за избраните от тях технологии и близък до закупения/ разработения от тях софтуер. Системните интегратори също не са маловажни. Те стоят между производителите на софтуер и операторите на инфраструктура и имат разнородни интереси. Например, ако се наложи подмяната на софтуера, който те са интегрирали, в определени случай това би донесло печалба (т.е. допълнителни поръчки за тях), а в други - загуби, ако операторът смени не само продукта, но и интегратора.

Тази разнородна смесица интереси на различни заинтересовани лица е довела до оформянето на три основни течения, определящи как трябва да бъде реализиран NGOSS – OSS/J (OSS through Java), MTOSI (Multi Technology Operation System Interface) и 3GPP.

2.4.1 OSS/J

OSS through Java стартира паралелно с NGOSS, като началото е поставено през 2000-та година. Целта на OSS/J е подобна на тази на NGOSS, като разликите са, че NGOSS е фокусирана основно върху бизнес и системните аспекти на OSS/BSS, а OSS/J върху създаването и внедряването на отворен OSS на базата на основни Java базирани технологии. С годините става ясно, че двете програми взаимно се допълват. NGOSS създава унифициран модел на бизнес процесите, system framework и UML информационни модели. OSS/J, следвайки принципите на NGOSS, редуцира SID модела до няколко

референтни модела наречени – CBE (Core Business Entities), и създава референтни отворени API (Application Programming Interface) интерфейси. API интерфейсите реално предоставят достъп до CBE през SOAP (Simple Object Access Protocol) базирана уеб услуга. Достъпът позволява основните CRUD (Create, Read, Update, Delete) операции да бъдат изпълнени върху обектите, създадени на база на CBE.

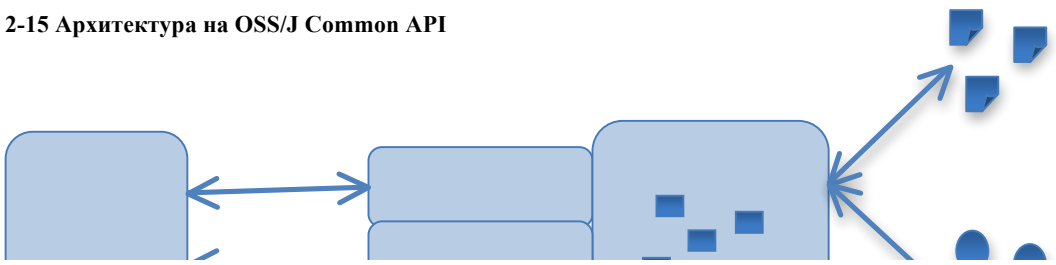
OSS/J специфицира следните отворени API интерфейси:

- Common - съдържа API общо за всички останали интерфейси. API-то предоставя JMS (Java Message Service) и XML (Extensible Markup Language) фасада на останалите интерфейси [71].
- Inventory Management - предоставя интерфейси за управление на CBE на три нива. Това са управление на клиенти, управление на услуги и управление на ресурси [72].
- Discovery – съдържа набор от интерфейси за управление на процеса по разкриване на мрежовите инфраструктури [73].
- Fault Management – съдържа набор от интерфейси за управление на процеса за управление на повреди [74].
- Order Management – съдържа набор от интерфейси за управление на поръчки [75].
- Trouble Ticket – съдържа набор от интерфейси на система за регистриране на повреди. [76].

OSS Application
Other OSS Applications
OSS Client
SOAP
Web Services Endpoint
WS Notification Producer
Managed Entities
Managed Resources

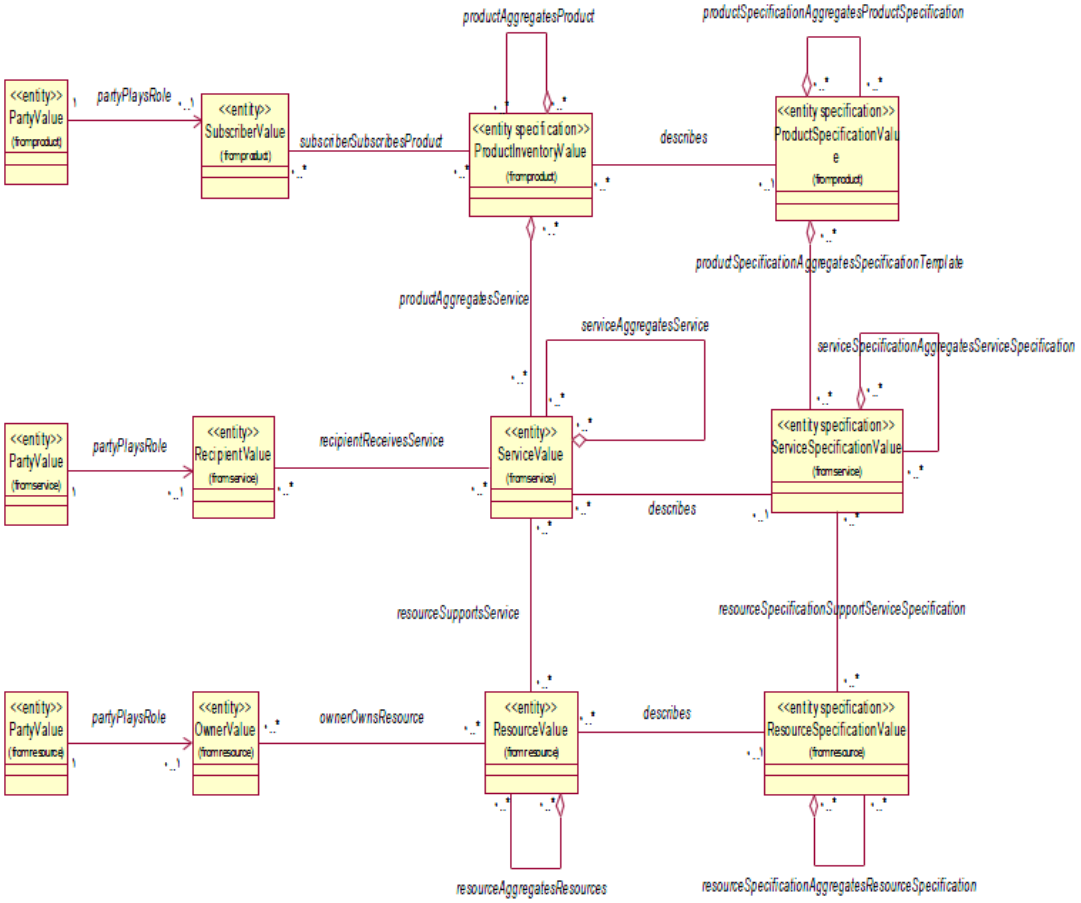
Имайки основните API интерфейси като база, разработчиците на JAVA OSS приложения могат да предложат унифицирана архитектура на OSS система. Предложеният модел акцентира върху използването на уеб услуги, чрез които дадената платформа би могла да бъде използвана от всяка една друга OSS (представена като OSS client). Същевременно платформата може на свой ред да има достъп до услугите предоставени от външни OSS приложения (Other OSS Applications) и да управлява определени ресурси (Managed Resources).

Фигура 2-15 Архитектура на OSS/J Common API



Статичният изглед към основния CBE модел, част от мета-модела на OSS/J Inventory API, е показана на Фигура 2-16. Той представлява базовия модел, който основните заинтересовани лица ще трябва да разширяват при всеки конкретен продукт и при внедряването във всяка една конкретна мрежова инфраструктура.

Фигура 2-16 Основни CBE в OSS/J Inventory API



CBE модела добавя към семантиката на UML следните понятия:

- Entity със стереотип “Entity” (<<Entity >>). Това е основна логически единица, представяща концепции като „Product”, „Service“ и „Resource”. Всяка логическа единица съдържа списък от методи, дефиниращи основните CRUD операции върху стойностите на описващите я атрибути.
- Entity спецификация със стереотип “Specification” (<<Specification >>). Това е логическа единица, специфицираща основните логически единици. Спецификацията съдържа характеристиките на основните логически единици.
- Асоциация със стереотип “Association” (<<Association >>).

OSS/J залага на базов модел и архитектура за достъп и управление на данните от модела, базирана на отворени интерфейси. Моделът на данните е общ, не специфицира

конкретни мрежови технологии и трябва да се разширява от всеки един продукт и за всеки един оператор. Това прави OSS/J подходящ за реализация на проекти, при които изискванията не са ясно специфицирани и е необходима архитектура, която да е гъвкава към динамично променящата се мрежова среда.

2.4.2 MTOSI

MTOSI предоставя независим от технологията набор от интерфейси за управление на телекомуникационни мрежи. В основата на MTOSI е модел на данните, специфициран в MTNM (Multi-Technology Network Management). MTNM е технологично ориентиран модел, повлиян основно от експерти с опит в мрежовите транспортни технологии. Основната цел е била да бъде създаден интерфейс от OS (Operation System) към OS и от NML (Network Management Layer) към EML (Element Management Layer) за управление на мрежи, използващи една или повече от една технология за пренос (SONET, SDH, PDH, ATM, Frame Relay, DSL, Ethernet). Интерфейсите са различни в зависимост от зоната на приложение. Примери за зони на приложение са управление на връзките, управление на конфигурацията, управление на модел на мрежата, управление на процеси по разкриване на мрежата и услугите в нея, управление на определени типове оборудване, управление на повреди и др.

MTNM първоначално се е използвал основно в приложения базирани на CORBA. Съответно интерфейсите са били специфицирани в CORBA IDL (Interface Description Language). Това е продължило до MTNM версия 3.5, когато няколко компании за пръв път са конвертирали CORBA IDL-и в XML. Това поставя началото на MTOSI.

Според дефиницията, текущата MTOSI 2.1 специфицира MRI (Manage Resource Inventory) интерфейс със следните възможности [77]:

1. Управление на ресурси:

- Resource Trouble Management (Управление на повредите, свързани с ресурсите) - управление на повреди, управление на резервираността, поддръжка и диагностика, управление на аларми;
- Managed Resource Inventory (Управление на ресурси) - управление на информацията, свързана с ресурсите на мрежата;

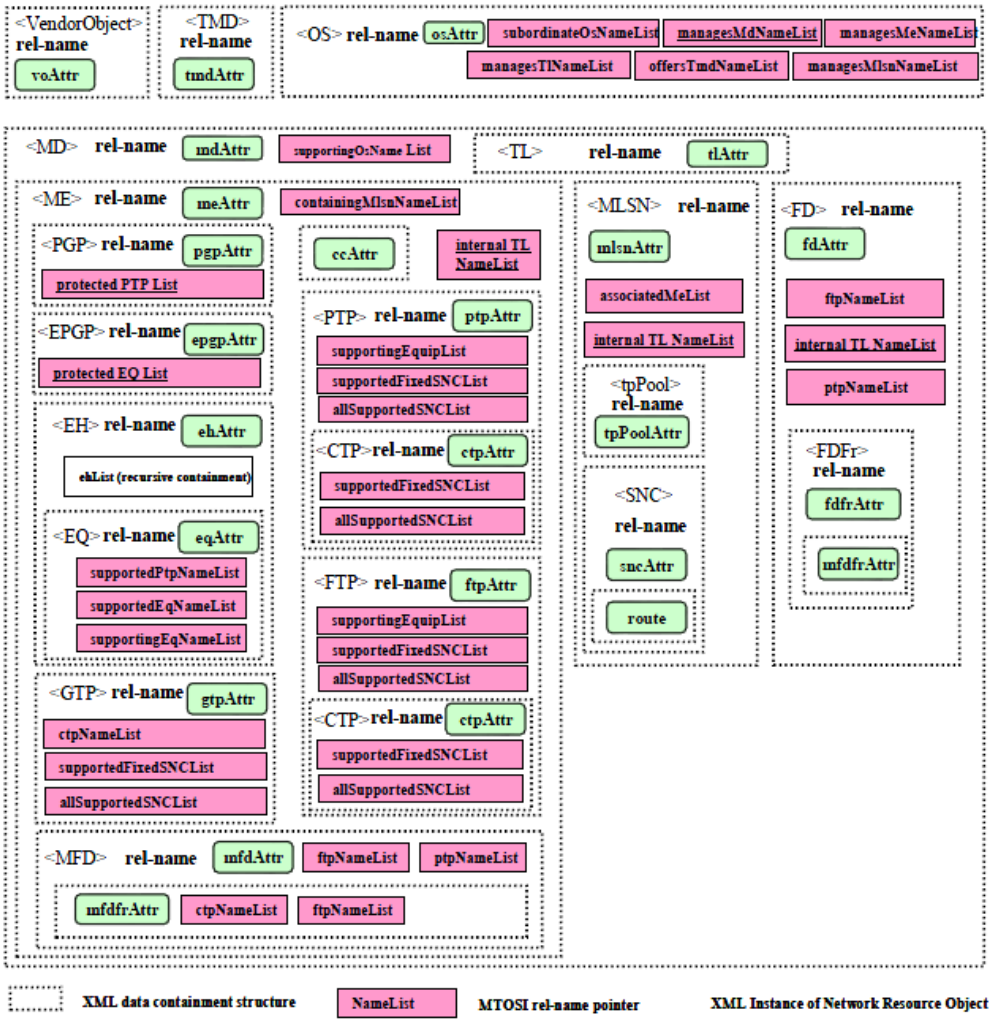
- Управление на ресурси – създаване, промяна и изтриване на ресурси от мрежата;
- Управление на производителността на ресурсите – наблюдение на ресурсите, генериране на аларми при достигане на определени стойности.

2. Управление на услуги:

- Активация и Конфигурация, включително активация между CRM OS и Service Configuration & Activation OS и между две Service Configuration & Activation OS;
- Управление на модела от данни – извличане на информация от модела, обновяване, добавяне и изтриване на данни.

Достъпът до интерфейса е през SOAP, използва WSDL – Web Service Definition Language, или през HTTP, или JMX. Първоначалният модел от данни, който се използва в MTOSI е базиран на MTNM, а не на SID. SID до версия 8 не съдържа логически единици, свързани с конкретни мрежови технологии. От версия 8 до текущата версия, SID моделът се разширява с все повече и повече MTOSI обекти.

Фигура 2-17 MTOSI XML inventory layout (xsd)



MTOSI използва XML XSD (XML Schema Definition) спецификации за дефиниране на референтните модели (Фигура 2-17). Диаграмите съдържат класове, атрибути на класовете и асоциации, представящи взаимовръзките между тях.

2.4.3 Сравнение между MTOSI и OSS/J

MTOSI е фокусиран в предоставянето на интерфейси на ниво услуга, мрежа и устройство. OSS/J от друга страна цели да предостави отворени интерфейси из целия OSS/BSS спектър. OSS/J интерфейсите не зависят от мрежовите технологии, докато тези предоставени от MTOSI зависят. През последните години тенденцията е OSS/J и MTOSI да бъдат обединени в един общ стандарт, но това все още не е факт.

В заключение може да се спомене, че и двете групи стандарти се стремят да интегрират различните OSS/BSS решения в една обща рамка. Този процес има множество предимства, но и голямо количество недостатъци. Реално интеграцията води до усложняване на продуктите и моделите на базата, на които работят приложенията от TAM. В стандартите и на OSS/J, и на MTOSI никъде не се споменава, как те влияят на системните свойства на OSS/BSS приложенията - не се коментира какво е отражението върху производителността, надеждността или как да се постигне определено ниво на сигурност. Самите стандарти представляват технологични обединения (technological frameworks) от множество интереси и не са подходящи за директна имплементация.

2.5 Адаптиране на системите за управление на мрежата и бизнеса към IPv6

В предходните подточки на настоящата глава бе разгледан модела на системите за управление на бизнеса и мрежовата инфраструктура на телекомуникационните оператори. Той се състои от няколко нива, като всяко едно ниво се състои от един или повече компоненти, изпълняващи определени функции. В крайна сметка функциите биват имплементирани в конкретни софтуерни системи, част от структура „Приложения“.

От гледна точка на прехода от IPv4 към IPv6, промените ще бъдат основно в eTOM ниво 2 или така наречения FAB модел. Fullfillment слоя ще трябва да започне да създава услуги за клиенти, базирани на IPv6. Системите за инвентаризация на мрежата ще трябва да бъдат променени, така че IPv6 да бъде включен по начин, подобен на сегашното представяне на IPv4. Системите за наблюдение на мрежовия трафик ще трябва да започнат да наблюдават и трафика, генериран от услуги и абонати в IPv6, а системите за таксуване ще се наложи да започнат да таксуват и услугите свързани с IPv6.

Адаптацията на бизнес процеса към IPv6 вероятно ще бъде по-трудна и бавна от адаптацията на съществуващите мрежи към IPv6. Процесът ще отнеме време и ще се отрази и на съществуващите системи в частта им, ориентирана към управление на бизнес процеса.

На първо място операторът ще трябва да започне да предлага услуги, базирани на IPv6. В първите години от наличието на IPv6, маркетинг отделите на големите доставчици

търсеха така наречените “killer applications“ - приложения, които да им дадат ключово предимство и на базата, на които да оценят прехода към IPv6. Подобни приложения разбира се не се появиха. IPv4, а също и IPv6 са основата на съвременните мрежови технологии. Самите те няма как да са “killer application”, но те са базата за въвеждането на подобни услуги. Примери за услуги, които биха имали големи предимства от въвеждането на IPv6 са така наречените облачни - “cloud” услуги и "Интелигентен дом”.

Облачните услуги предполагат наличието на голям брой машини и приложения, работещи в общи layer 2 сегменти и предоставящи услуги, видими от Интернет. С други думи голям брой крайни устройства с реални IP адреси. IPv4 разчита на Address Resolution Protocol (ARP), за да изгради препредаващите таблици в layer 2 комутаторите. ARP работи на базата на ”broadcast” – механизъм, неподходящ за среди с голям брой устройства. От друга страна IPv6 използва механизми за откриване на най-близкия маршрутизатор и не използва ARP. Broadcast механизъма също е заменен с мултикаст. Тези две промени водят до значително увеличение на броя сървъри в един мрежови сегмент и дават съществено предимството на облачните технологии, базирани на IPv6.

Напълно реална е възможността услугите, базирани на IPv6 да имат по-ниска себестойност от тези върху IPv4. Налице са първите сделки, при които доставчици на Интернет и облачни услуги инвестираха в големи префикси от IPv4 адресно пространство. Това автоматично означава, че услугите предоставяни от тях върху IPv4 ще бъдат натоварени с допълнителните разходи свързани с все по ценните IPv4 адреси.

Например Microsoft закупиха 666,624 IPv4 адреси от фалиралия Nortel за скромната сума от $7.5 mln. Това прави по $11.25 на адрес. Проблем има, адресите свършват и скоро ще има диверсификация на цените на услугите предлагани върху IPv6 и IPv4. Подобни примери ясно показват, че услуги, изискващи значителен брой реални IP адреси ще бъдат по-евтини, ако бъдат предоставени върху IPv6.

За да станат реалност подобни предложения, от една страна ще трябва да еволюира не само мрежата на доставчиците, но и използваните системи за нейното управление, а също и системите за управление на бизнеса. Например системата за създаване на абонати и услуги ще трябва да се адаптира към конфигурационните промени, свързани с новия протокол. Центровете за обслужване на клиенти ще трябва да започнат да обработват

заявки от клиенти от IPv6 домейна, а системите за управление на взаимоотношенията с клиента ще трябва да отразят дали дадения клиент използва IPv4 или IPv6 услуги. Не е изключено за в бъдеще да започнат да се предлагат и услуги с една цена, ако са реализирани върху IPv4 и с друга, ако са IPv6.

Основната цел на настоящата докторантура е да предложи подход за еволюция на мрежата от текущото IPv4 състояние до желано такова, характеризиращо се изцяло с IPv6. Подобна еволюция би била невъзможна без софтуер подобен на този, специфициран в OSS/BSS стандартните. Същевременно съвременният OSS/BSS е създаден, за да управлява, а не да еволюира дадена мрежа. Процесът на развитие на OSS/BSS следва мрежовата инфраструктура. Процесът на еволюция налага създаването на софтуер способен да моделира и еволюира дадена мрежа. Интеграцията между софтуера за еволюция на дадена мрежа и софтуера за управлението ѝ може да стане на ниво модел на данни. Например, данните, генерирани от системата за еволюция, могат да допълнят съществуващите данни в SID модела, използван в OSS/BSS.

2.6 Основни изводи в резултат от направения анализ на системите за управление на мрежите и бизнеса

Мрежовите оператори използват различни OSS/BSS системи, чиято основна цел е постигане на ефективно управление на мрежата и извличане на максимален доход от съществуващата инфраструктура. В детайли са разгледани три от основните сфери на приложение на OSS/BSS – процесите по осъществяване на поръчки, осигуряване на мрежата и събиране на приходи от нея.

Детайлно са представени стандартите и спецификациите, които представят модели на отделните процеси, съставящи OSS/BSS системите. Два от най-широко използваните модели са ITU-T TMN и NGOSS eTOM. Всеки от тях представя групиране на процеси на различни нива според областта им на приложение и дефинира съставни подпроцеси. Основната цел е да се създаде универсален модел на протичащите бизнес процеси и взаимодействието между тях в мрежови оператор, доставчик на услуги и търговец на услуги.

Протичането на тези процеси е съпроводено с използване и обмяна на различна по вид информация. Поради тази причина се появява необходимостта от дефиниране на универсален информационен модел. Такъв е дефиниран от NGOSS и се нарича интегриран информационен модел (SID). Неговата структура следва различните нива на бизнес процесите в eTOM.

Бизнес процесите и обменяната между тях информация се обединяват в приложения за управление на клиенти, услуги и ресурси. Моделът на приложенията е йерархичен и е дефиниран от TAM.

Практическото реализиране на дефинираните модели на процеси, информация и приложения не е лесно поради различните заинтересовани лица. Поради тази причина възникват три основни течения, които определят как трябва да бъде реализиран NGOSS – OSS/J, MTOSI и 3GPP.

В тази глава е направен:

- обзор и сравнение на модели за описание на OSS/BSS процеси;
- обзор и сравнение на стандарти за реализиране на NGOSS;
- анализ на необходимите промени на eTOM ниво 2 (FAB модел) при реализиране на преход от IPv4 към IPv6.

В заключение е анализиран OSS/BSS по отношение на процеса на еволюция на мрежата и в частност относно прехода от IPv4 към IPv6. Според автора OSS/BSS не е създаден с цел еволюция на мрежата, а с цел нейното управление. Това предполага, че ако бъде създадена функционалност, която да еволюира дадена мрежа, то тя трябва да бъде интегрирана с OSS/BSS, така че системите за управление да имат актуална информация за модела на текущата мрежа. Интеграцията може да стане на ниво SID модел, като данните за достигнатото еволюционно ниво бъдат попълвани в интегрирания SID модел на OSS/BSS.

Глава 3: Подход и предложение за решаване на проблема

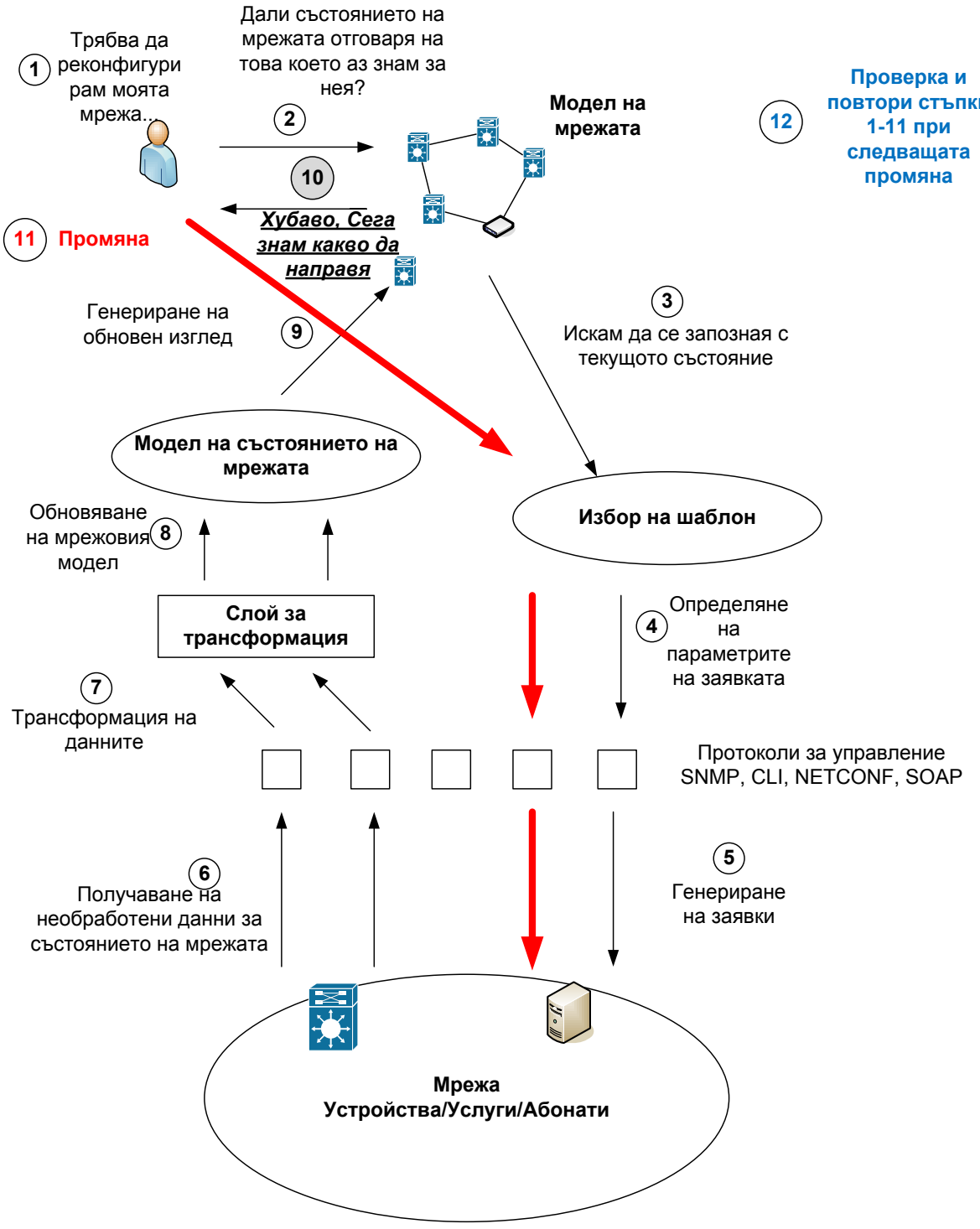
В настоящата глава ще бъде представен подходът на автора за решаване на проблема по трансформация на мрежи и услуги от IPv4 към IPv6.

3.1 Основна идея

Изходната идея, която покрива функционалните изисквания към прототипа за преход от IPv4 към IPv6 е представена като поредица от логически действия (1 – 12) на Фигура 3-1.

1. Мрежата трябва да отговори на дадена промяна в заобикалящата я среда. Промяната може да бъде въвеждане на нова услуга, добавяне на нов абонат или реакция в резултат от настъпването на дадена повреда. В голяма част от случаите промяната се изразява в конфигурация на ново или съществуващо устройство или пък в добавяне или подмяна на оборудване.
2. Преди да бъде направена каквато и да е промяна в „жива“ мрежа е добре да бъде зададен въпроса дали познанията на инженера за нея са актуални. Познанието в голяма част от случаите се свързва с визията за дадена топология. За да бъде изградена топологията е необходимо да бъде извлечена информация от мрежата по даден протокол.
3. Командите за извличане на информация или за промяна състоянието на мрежата могат да бъдат описани в шаблони. Шаблоните съдържат комбинации от заявки и се използват за описване на поредиците от заявки, които инженерите изпълняват .

Фигура 3-1Основна идея



4. За да бъдат изпълнени командите от шаблона, трябва да им бъдат подадени набор от входящи параметри.

5. Изпълнението на заявките става чрез протоколи за управление на мрежови устройства като SNMP, NETCONF, CLI (Telnet/SSH).
6. Мрежата връща необработени данни в отговор на заявките. Необработените данни трябва да бъдат нормализирани.
7. Нормализацията става чрез трансформация на необработените данни във формат, удобен за визуализация на мрежовата топология.
8. Новите данни обновяват модела на мрежата.
9. На базата на обновения модел се визуализира обновената топология на мрежата.
10. Топологията и промените в нея биват анализирани.
11. Вече може да бъде взето решение какво и как да бъде променено в мрежовата инфраструктура.
12. Провери резултата и ако е успешен повтори стъпки от 1 до 11 за следващата промяна

В течение на работата по настоящата теза бе установено, че описаната поредица от действия се повтаря ден след ден във всяка една мрежа. Независимо дали се извършва миграция към IPv6 или се въвежда в експлоатация нова услуга, поредицата от действия си остава реално една и съща. Тя винаги се състои от някакво действие, което може да бъде изпълнено при определени условия и което ще има определен ефект върху инфраструктурата. Действието винаги носи определен риск, отнема време и си има цена. Комбинацията от всичките тези елементи може да се определи като стъпка, която променя текущото състояние на мрежата.

Възниква въпросът дали процеса на непрекъсната промяна на състоянието на мрежата не може да бъде систематизиран, така че да обедини техническите действия с изискванията на бизнеса. Бизнес изискванията се превръщат в стратегия за развитие на дадената компания. Стратегията отговаря на изискванията на редица вътрешни и външни за дадената компания заинтересовани лица. Тези изисквания биват породени от редица обстоятелства, свързани с екосистемата, в която се развива дадения бизнес. В крайна сметка изискванията на бизнеса могат да бъдат реализирани чрез поредици от множество технически стъпки. Стъпките могат да бъдат групирани в стратегии. Заинтересованите

лица трябва да имат начин да оценят отделните стратегии за развитие и да изберат най-подходящата за развитие на техния бизнес.

3.2 Дефиниция на проблема

За да се дефинира проблема, ще бъде представен конкретен хипотетичен пример с доставчик на Интернет услуги - "X". Представения описание е дефинирано на базата на контекста на реален мрежови оператор, върху чиято мрежа автора е направил изследване на потенциалните начини за въвеждане на IPv6 [10]. Освен това са използвани и принципите на моделиране на интересите на различни заинтересовани лица дефинирани в [78] и [58].

3.2.1 Контекст и заинтересновани лица

Операторът X е среден по размер доставчик на Интернет услуги с около 500 бизнес клиенти и 50000 домашни абонати. За напред оператор X ще е просто X. X предоставя на корпоративните си клиенти, мрежови услуги като MPLS VPN мрежи, хостинг, Интернет, телефония и редица други услуги с добавена стойност. Примери за последните са качество на обслужване, сигурност, корпоративен интернет достъп, услуги по управление на клиентската мрежова инфраструктура и др. На домашните си абонати, X предоставя услуги като Интернет, телефония, телевизия и услуги с добавена стойност като "pay per view" и "video on demand".

X е компания с десет годишен опит и се представя добре на пазара на телекомуникационни услуги. През всичките тези години маркетинг отдела на компанията е успявал да намери верния път и да изкара на пазара точните продукти в правилните моменти. Отдел "Продажби" също се справя добре. X не само, че успява да запази текущите си клиенти, но и чрез последователна политика за разширение на мрежата добавя по 5% нови абонати на година.

Моментът е ключов за X. Пазарът е наситен откъм конкуренти, подбиващи цената на традиционните предложения на X - корпоративната свързаност и достъпа до Интернет. Ситуация близка до тази на българския телекомуникационен пазар. Висшия мениджмънт на X е подложен на огромен натиск от акционерите на компанията да намери правилния път и да задържи нивото на печалба от предходните години.

Маркетинг отделът на компанията има няколко предложения, чрез които да изведе X на нови пазари и не само да удържи на натиска на конкуренцията, но и да я постави в позиция на водеща компания с години напред. Предложенията се свеждат до две често срещани понятия в мрежовите среди - въвеждане на нови технологии и внимателно планирано разширение на мрежата в райони, където X все още не присъства.

Отдел “Продажби” нетърпеливо чака момента, в който ще може да предложи продукти, базирани на новите технологии, а също така и възможността да подбие цените на конкурентите си в районите където X все още не присъства.

Отдел “Развитие на мрежата” е поставен под огромен натиск от ръководството на компанията да предложи печеливша стратегия за въвеждане на новите технологии и контролирано разширение в новите райони, при това за максимално кратък срок на минимална цена. Мрежовият архитект на оператор X се опитва да дефинира различни стратегии за еволюция на мрежовата инфраструктура според текущото състояние на мрежата, изискванията на новите технологии и плановете за разширение.

Отдел “Поддръжка на мрежата” гледа тревожно на плановете на висшето ръководство на компанията и на “безумията”, които предлагат от отдел ”Развитие на мрежата”. В крайна сметка на хората от отдела им е ясно, че всяка промяна носи риск и има определена цена. Според договорните отношения със съществуващите клиенти, всяко прекъсване на мрежата извън регламентираните за целта “прозорци” се заплаща скъпо от оператора. Отдел “Поддръжка на мрежата” разполага с много кратки интервали от време, за да изпълни планираните промени.

Интересът на отдел „Сигурност“ е нивото на сигурност на мрежовата инфраструктура да остане същото или да стане по-високо в сравнение с досегашното ниво.

3.2.2 Изисквания

Предложенията на отдел “Маркетинг” се свеждат до въвеждането на две нови технологии - Internet of Things [79] и Cloud [80].

3.2.2.1 Internet of Things

Първото предложение е за предлагане на решения базирани на ранни „Internet of Things" решения за автоматизация на домове на домашни абонати и сгради на корпоративни клиенти. Решението ще е комбинирана оферта от страна на оператора и външни компании, специализирани в предлагането на интелигентни решения за автоматизация на сгради. Основната цел е във всяка сграда да бъде инсталирано ново устройство наречено home gateway. Чрез него могат да бъдат свързани всички интелигентни устройства в сградата на клиента. Примери за такива са: интелигентни хладилници, решения за видео наблюдение, телевизори, печки, парно, системи за сигурност, датчици за пожар и наводнение, интелигентни бушони и много други устройства. От страна на оператора се изисква да бъде предоставена надеждна свързаност между новото устройство и системите за управление на интелигентните устройства. Колкото до самите системи, всяка външна компания, предлагаща такива, ще може да сключи споразумение с оператора и да ги инсталира в частния "облак", предлаган от оператора. Услугите разбира се може да се предоставят и от публични облаци - част от Интернет, но тогава операторът не поема никаква отговорност за трафика между облака и Интернет, и не гарантира качество по-различно от това на стандартната Интернет услуга.

3.2.2.2 Частен Облак (Private Cloud)

Операторът X иска да се присъедини към пазара на една от най-дискутираните напоследък услуги, а именно предлагането на всевъзможни "облачни" услуги. Стратегията на отдел "Маркетинг" е да се започне с предлагането на частни "облачни" предложения за корпоративни клиенти и външни доставчици на услуги върху инфраструктура от виртуални машини. Като хардуер, първоначално решението е обезпечено от част от досегашните сървъри, използвани за предлагането на недобре продаваните хостинг решения.

3.2.2.3 Разширения

В бизнес плана на компанията е декларирано, че през следващата година тя трябва разшири обхвата на своята мрежа в над четири града и десет квартала. Това ще позволи достъп до повече от пет хиляди нови абонати. За целта трябва да бъдат добавени четири

опорни маршрутизатора (Provider - P routers), десет маршрутизатора за обединение на трафик (Metro Provider Edge - Routers) и над десет крайни комутатора. По възможност новите устройства трябва да са от моделите, сертифицирани за работа с досегашната мрежа.

3.2.2.4 Подмяна на старо оборудване

На различни възлови точки в сегашната мрежа има оборудване, което е старо и все по-трудно удържа на нарасналите трафични натоварвания. Част от него вече не се поддържа от страна на производителите и за него не се предлагат нови версии на софтуер или подмяна на хардуерни елементи. Отделът "Поддръжка на мрежата" има бюджет да замени най-критичните подобни устройства с нови. Веднъж инсталирани, новите устройства трябва да бъдат свързани към мрежата по същия начин, като старите и да поддържат всички услуги, работещи върху досегашните рутери.

3.2.3 Ограничения и качествени характеристики

Операторът X трябва да осигури качествени мрежови услуги на своите клиенти. Качеството обикновено означава гарантиране на определени трафични параметри (забавяния, jitter, капацитет) за определени видове трафик и с определена надеждност (99.99% или 99.999%). Операторът X може да прави промени, засягащи трафика на корпоративните клиенти само в ясно дефинирани периоди от време, описани в договорите с тях. Операторът трябва да ги уведомява минимум 2 дни предварително. Традиционно тези периоди са в часовете на минимално натоварване на мрежата. Проблемът е, че такива почти няма. Вечер и през почивните дни са активни домашните потребители, а през деня в работно време корпоративните. Корпоративните клиенти също така използват нощните часове, за да синхронизират големи масиви от данни в периоди, когато това не би засегнало основната им бизнес дейност. Най-платежоспособните клиенти имат клаузи в договорите си, че периодът не трябва да надвишава един час във време, с което самите те са съгласни. Всяко отклонение от описаните норми се наказва с плащане на наказателни такси от страна на оператора към неговите клиенти.

3.2.4 Анализ на изискванията

Изискванията описват ситуацията във виртуалния мрежови оператор X. Въпреки, че операторът е виртуален тези изисквания са доста близки до действителността и до задачата, която трябва да бъде разрешена от много съвременни доставчици на мрежова свързаност и Интернет услуги. Във всяка една от тези компании има мрежови архитект или дори екипи от такива, опитващи се да анализират ситуацията и да представят на висшите управленски нива различни стратегии за това как да разрешат пъзела, свързан с предлагане на конкурентни услуги.

Колкото до самите висши управленски нива, те обикновено са съставени от различни хора, ръководещи различни отдели, с различен начин на мислене, които взимат решение на базата на различни комбинации от критерии. Например ръководителят на отдел "Поддръжка на мрежата" би се водил основно от това да минимизира риска, някои други ще държат единствено на разходите, а трети ще искат минимално кратки срокове за въвеждането в експлоатация. Наличието на различни мнения сред висшите управленски кадри на една голяма организация, каквато е традиционния телекомуникационен оператор или средните и по-големи Интернет доставчици, е често срещан случай. Важното е всеки един с право на глас да има достатъчно точни аргументи на базата, на които да вземе решение. Аргументите трябва да дойдат от мрежовия архитект и той е човека, който трябва да подбере правилните аргументи и перспективи на мрежата.

Като заключение на казаното по-горе, мрежовите архитекти трябва да могат да анализират добре ситуацията в компанията, текущата инфраструктура, пазарът на услуги, конкуренцията и новите технологиите. На базата на тази информация те трябва да представят на хората, натоварени с вземането на решения, различни стратегии за това, как дадената мрежова инфраструктура ще еволюира. Ръководителите на компанията от своя страна вече ще изберат, коя стратегия е най-подходяща спрямо целите и ограниченията, с които трябва да се съобразява дадената компания. Обикновено ограниченията може да бъдат най-просто изразени като време, цена и риск.

3.3 Подход на автора

Преходът на дадена мрежова инфраструктура от IPv4 към IPv6 може да бъде описан като преход от едно текущо (IPv4) състояние на мрежата към друго желано (IPv6). За да

бъде достигнато желаното състояние, мрежата ще трябва да премине през едно или повече от едно преходни състояния. Преходите между тези състояния може да се определят като стъпки, които трябва да се изпълнят, за да се реализират дадени промени. Стъпките може да бъдат групирани в стратегии. Достигането на желаното състояние може да стане чрез изпълнението на една или друга стратегия. Пътят, по който това състояние ще бъде достигнато, ще се нарича еволюционен.

3.3.1 Състояние

Всяка мрежа във всеки даден момент е в дадено състояние. Състоянието може да бъде описано на базата на множество статични и динамични параметри. Например сред статичните параметри са хардуера на устройствата, софтуера, връзките между възлите, конфигурацията на устройствата, а сред динамичните са текущите характеристики на протоколите, статуса на устройствата и интерфейсите.

Допускане 1: Текущото състояние може да бъде представено като граф, състоящ се от възли и ребра с дадени параметри, изразяващи статични или динамични свойства на дадена мрежа.

3.3.2 Граф

Всеки граф може да бъде изразен като G = (V,E), където V = {v1, . . . , vn} е множество от възли, а E = {e1, . . . , em} е множество от ребра (връзки) между възлите.

3.3.3 Възли

- Всяко едно устройство, налично в дадена мрежа е възел.
- Всеки възел има уникален идентификатор (ID) и се характеризира с множество от свойства.

3.3.4 Ребра

- Всяка връзка между два възела е ребро.
- Всяко ребро има уникален идентификатор (ID), връх, от който излиза (source) и връх, до който достига (target). Всяко едно ребро се характеризира с множество от свойства.

3.3.5 Модел на графа

Графът, отговарящ на дадено състояние на мрежата може да бъде съхраняван в различни модели от данни. Форматът, използван от автора е Graphml [81]. Опростен граф, изразен в Graphml формат е представен на Фигура 3-2.

Фигура 3-2 Graphml формат

```
<graphml>
  <graph edgedefault="directed">
     <key id="weight" for="edge" attr.name="weight" attr.type="double"/>
     <key id="property" for="node" attr.name="name" attr.type="String"/>
    <node id="v1">
       <data key="color">green</data>
    </node>
    <node id="v2">
       <data key="color">green</data>
    </node>
    <node id="v3"/>
    <node id="v4"/>
    <edge id=”v1v2” source="v1" target="v2">
         <data key="weight">1.1</data>
    <edge id=”v1v3” source="v1" target="v3"/>
    <edge id=”v1v4” source="v2" target="v4" directed="false"/>
  </graph>
</graphml>
```

Моделът може да се тълкува по следния начин. Съществува граф, който съдържа възли с идентификатори v1, v2, v3 и v4. Всеки възел е представен с отделен xml <node> елемент и съдържа под себе си определено количество свойства, описани като xml <data> елементи. Всеки <data> елемент съдържа xml key атрибут, описващ името на свойството. Например възел v1 има свойството color (цвят) “green”. По същия начин, графът съдържа ребра между възлите. Всяко ребро започва от един възел и завършва в друг. Свойствата на ребрата са изразени отново с <data> елементи. Името на елемента се съдържа в xml key атрибут. Например ребро v1v2 започва от възел v1 и завършва във възел v2. То има свойството weight (тежест) със стойност 1.1.

3.3.6 Трансформация на мрежата от едно състояние в друго

Допускане 2: Процесът на мрежова трансформация може да бъде формално изразен и описан, като промяна на модела на мрежовия граф от текущо състояние към бъдещо „желано“ състояние.

Всяко едно от състоянията може да бъде формално описано в Graphml. Първоначалното състояние на мрежата може да съдържа даден списък от възли и връзки с дадени свойства, а крайното състояние може да съдържа различен списък от възли и връзки. Всеки възел/ връзка може да претърпи промяна на стойностите на свойствата си между началното и крайното си състояние, а също така е възможно да бъдат добавени или изтрити дадени свойства.

Трансформацията между две състояния може да се изрази като трето състояние, съдържащо общите възли и връзки, и допълнителна информация за разликите между двете състояния.

3.3.7 Команди

Допускане 3: Промяната на свойствата ще става на базата на действия. Действията може да съдържат една или повече команди. Командите може да бъдат групирани в шаблони.

Примери за подобни команди са тези, необходими за вход, автентикация и изход в командния интерфейс на дадено устройство. Командите може да са свързани с промяна на кой да е параметър на устройството, с рестартиране или пък зареждане на нов софтуер. Командите се нуждаят от правилно подадени параметри, за да доведат до желан резултат. На Фигура 3-3 е демонстрирана команда, която конфигурира IP адрес **$ipAddress** и маска **$subnetMask**. **$ipAddress** и **$subnetMask** са входящите параметри, необходими за изпълнение на командата.

Фигура 3-3 Примерна команда

```
ip address $ipAddress $subnetMask
```

3.3.8 Шаблони

Шаблонът дефинира последователност от команди, условията, при които те трябва да се изпълнят и нужните им параметри (Фигура 3-4). За целта са дефинирани следните ключови думи:

###vars - Дефинира параметрите, които шаблона трябва да получи. Всеки един параметър се замества в една или повече команди по начина, показан на Фигура 3-4.

###read_until - Гарантира, че дадена команда няма да бъде изпълнена преди да бъде получен определен отговор от устройството.

start_read_until - Улеснява дефиницията на поредица от команди преди всяка, от които трябва да бъде получен един и същ отговор от устройството. Например на Фигура 3-4 това е идентификатора на командния интерфейс –“#”.

###exit - Край на шаблона и край на сесията с мрежовото устройство.

Фигура 3-4 Шаблон

```
### vars: username, password, firstFreeInterface, ipAddress, subnetMask
### read_until('(login:|user:|Username:)',3)
$username
### read_until('(Password:|password:)',3)
$password
### start read_until('.*#',3)
set cli  screen-length 0
configure terminal
intrface $firstFreeInterface
ip address $ipAddress $subnetMask
no shutdown
### stop read_until
exit
### exit
```

3.3.9 Стъпки

Допускане 4: Процесът на трансформация ще се случи на определен брой „архитектурно значими“ стъпки, докато бъде достигнато желаното състояние.

Допускане 5: Всяка една стъпка се състои от технически и бизнес ограничения, действия, които ще се изпълнят за дадено мрежово устройство и проверка дали действието реално е довело до очакваното състояние.

Примерната реализация на дадена стъпка може да се дефинира чрез псевдокод по следния начин.

Фигура 3-5 Формално описание на стъпка чрез псевдокод

```
$params = $param1, $param2, …
if(guard ($params)  eq 'YES'){
boolean $actionResult = action(params);
  if ($actionResult eq true){
            updateState(params);
            boolean $effectResult = effect($params);
            if ($effectResult eq true){
                        exit
            }else{
                        rollback($params);
            }
  }
           else{
            rollback();
  }
}
```

3.3.9.1 Технически ограничения

Техническите ограничения може формално да се изразят като предварителни условия, на които трябва да отговаря модела на мрежата преди да бъде предприето дадено действие в нея. Те предпазват мрежата от изпълнението на неправилни действия. Синтаксисът на език от команди и проверки, които трябва да бъдат направени преди тяхното изпълнение не е новост и е дефиниран от Едгар Дийкстра в [82]. Дийкстра дефинира "Guarded Command Languadge" език, според който задължителното условие – "guard" е твърдение, което трябва да е истина преди да бъде изпълнена дадена команда или команди. В системата за трансформация на мрежи от IPv4 към IPv6 "guard" е проверка по отношение на текущия модел на графа, която трябва да даде верен резултат. Това формално е изразено на Фигура 3-6. Смисълът на логическия оператор е следния - ако моделът съдържа възел с идентификатор R1 и той има свойството „ipv6Forwarding" със стойност „YES", да се изпълни следното действие. Проверката е дефинирана чрез XPATH [83] синтаксис спрямо графовидния graphml модел на мрежата.

Фигура 3-6 Guarded Command syntax

```
$ID=R1
If (XPATH(//graphml/graph/node[@id='$ID']/data[@key='ipv6Forwarding'])  eq 'YES'){
изпълни действие
}
```

3.3.9.2 Ограничения, свързани с бизнеса и организацията (Business Constraints)

Може да има различни видове ограничения, свързани с бизнеса и организацията, които да бъдат асоциирани с дадена стъпка. Бизнес ограниченията се изразяват като конкретни параметри, свързани с дадена стъпка.

По-долу са дадени примери за три от най-често срещаните такива - риск, цена и време.

Риск - рискът е фактор, който може да се определи като зависимост от Likelihood (вероятността изпълнението на стъпката да не е успешно) и Impact (последствията от този неуспех). Крайният риск може да бъде Note (никакъв), LOW (нисък), MEDIUM (среден), HIGH (висок) или Critical в зависимост от стойностите на скалите Likelihood и Impact.

Фигура 3-7 Скала за определяне на риска

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

- Рискът е Note, ако Likelihood и Impact са Low.
- Рискът е Low, ако едната от двете скали е Low, а другата Medium.
- Рискът е Medium, ако стойностите по двете скали са Medium или ако едната от двете скали е High, а другата е Low.
- Рискът е High, ако едната скала е High, а другата Medium.
- Рискът е Critical, ако и двете скали са High.

Предложената методология за определяне на нивото на риск не е задължителна и е използвана за илюстрация на подхода. Операторите на мрежова инфраструктура имат собствени методологии за определяне на риска и могат да ги използват вместо предложения от автора модел.

Цена – това е цената, на която ще бъде извършена дадената стъпка. Например това може да са разходи за нови устройства, разходи за подмяна на версията на софтуера, разходи за персонал и консултанти, разходи за обучения и дори разходи за IPv4 адреси.

Време – дадена стъпка може да бъде асоциирана с различни периоди от време. Например голяма част от мрежовите оператори нямат практика да правят определени промени по „живата“ си мрежа кога да е. Промените се извършват в специално определени интервали от време в часовете на най-малък трафик. В примерите по-долу, когато има времеви ограничения, те ще са във време част от тези интервали.

Стъпката, също така би могла да бъде асоциирана и с времена за подготовка, проучване и тестване на стъпката в лабораторни условия.

3.3.9.3 Действие (Action)

Действието се изразява в прилагането на даден шаблон или команда върху конкретно мрежово устройство, връзка или група от такива.

3.3.9.4 Проверка (Effect)

Практиката показва, че няма оператор, който да не държи на механизъм за проверка, дали дадено действие в мрежата е успешно или не. Според подхода на автора проверката може да бъде два типа:

- Проверка, дали обновения мрежови модел наистина се намира в очакваното състояние. Това се изразява в намиране на разликите между достигнатото състояние и предишното състояние и проверка, дали в разликите се съдържат очакваните промени. Технически това би могло да бъде осъществено чрез намиране на разликите между графовете на двете състояния, изразени в graphml формат. Резултатът от сравнението ще бъде трети граф, съдържащ допълнителни атрибути и свойства, изразяващи наличието на разлики.

Проверката на ефекта в крайна сметка може да се изрази като XPATH израз, изпълнен по отношение на резултата от сравнението на двата графа.

- Проверка в реалната мрежа - пример за това е изпълнението на команди за проверка на свързаността, като ping и traceroute.

Практиката е показала, че ако нещо ще се променя в жива мрежа, винаги трябва да се знае, как да се подходи в случай на неуспех. Трябва да бъде дефинирана и стъпка, която би върнала мрежата в предходното ѝ състояние. Тази стъпка в общия случай отново има гард, действие и най-вече ефект. Ефектът обикновено се изразява в проверки подобно на гарда на основната стъпка.

3.3.10 Стратегии

Допускане 6: Стъпките могат да се групират в стратегии. Стратегиите се отличават една от друга по стъпките, от които се състоят и по междинните състояния, през които ще премине мрежата при изпълнение на стратегията. Всяка една стратегия се характеризира с конкретни технически и бизнес ограничения според стъпките, от които е съставена.

Подборът на стъпките в дадена стратегия зависи основно от техническите и бизнес ограниченията.

3.3.10.1 Влияние на бизнес ограниченията

Бизнес ограниченията влияят на основната цел на стратегията и в общия случай, заедно с техническите ограничения, предопределят стъпките, от които тя ще се състои. Бизнес ограниченията влияят и генерално на подредбата на самите стъпки.

Например, ако даден оператор няма бюджет за капиталови разходи, свързани с подмяна на оборудване и разширения на мрежата, е възможно да осъществи процеса по въвеждане в експлоатация на дадена услуга максимално бързо. По този начин евентуалните приходи от услугата ще покрият капиталовите разходи, свързани с разширенията и подмяната на оборудване. Следователно, началото на стратегията ще се състои от стъпки, свързани с употреба на механизми по въвеждане на услугата в експлоатация, средата - от стъпки свързани с подмяна на оборудване и въвеждане на ново такова, а краят - от стъпки по премахване на механизмите въведени в началото.

Съответно, ако операторът има необходимия бюджет, може да извърши стъпките по подмяна на оборудване и разширяване на мрежата в началото на стратегията и след това да въведе в експлоатация новата услуга. По този начин би избегнал изпълнението на ненужни стъпки, свързани с отстраняването на временни механизми.

3.3.10.2 Влияние на техническите ограничения

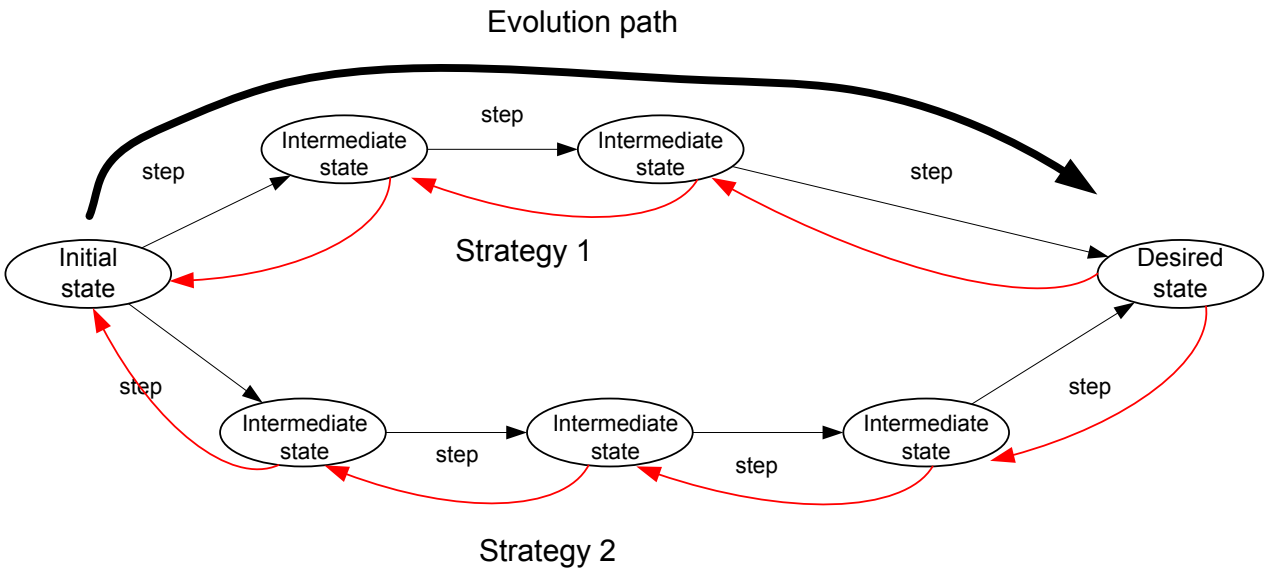
Техническите ограниченият оказват влияние върху конкретните стъпки, които биха били използвани в конкретна стратегия. Пример за подобни ограничения е забраната за използване на конкретен механизъм за преход от IPv4 към IPv6 в дадена мрежа.

Техническите ограничения също така оказват влияние върху точната последователност от стъпки в дадена стратегия. От тях зависи дали една стъпка може да се изпълни или не. Поради това стъпките в стратегията трябва да бъдат подредени в такъв ред, че тяхната изпълнимост да бъде предварително гарантирана.

3.3.11 Еволюционен път

Допускане 7: Еволюционният път е стратегията, отговаряща най-добре на изискванията на различните заинтересовани лица. По него мрежата еволюира от първоначалното си състояние до желаното такова.

Фигура 3-8 Еволюционен път, стратегии, стъпки и междинни състояния



Пътят от текущо (Initial State) до желано (Desired state) състояние преминава през множество от междинни състояния. Преходът между състоянията се определя от стъпките, част от одобрената стратегията. Мрежата еволюира по графовиден път. Пътят се състои от първоначално, крайно и множество от междинни състояния. Преходът между едно и друго състояние се извършва според стъпките, от които се състои еволюционния път.

3.3.12 Алгоритъм за определяне на еволюционния път

Еволюционният път бива избран на база на оценка на ограниченията, наложени от заинтересованите лица и условията, наложени от средата, в която оперира дадения оператор, в съответствие с ограниченията, асоциирани с конкретните стъпки.

3.3.12.1 Критерии за избор

Критериите за избор на еволюционен път се изразяват формално спрямо ограниченията, асоциирани с всяка една стъпка. Примери на критерии за избор на еволюционен път са:

- Минималното време за достигане на състояние, при което мрежата започва да поддържа дадена нова услуга (например „Интелигентен дом“). Критерият може да бъде изразен като сума от времената за изпълнение на стъпките до достигане на даденото състояние.
- Минималните разходите за достигане на конкретно състояние. Критерият може да бъде изразен като сумата от разходите свързани с изпълнение на стъпките до достигане на даденото състояние.
- Минималният максимален риск до достигане на дадено състояние. Например рискът за достигане на състоянието Интелигентен дом трябва да бъде по-малък или равен на High.
- Минимален осреднен риск. Критерият може да бъде изразен формално като минималната средна стойност на осреднения риск от стъпките до достигане на дадено състояние.

- Механизмите за преход чрез превод на адреси не са разрешени. Критерият може да бъде изразен като техническо ограничение на ниво стратегия със стойност “nat”’=NO.
- Двойният IP стек не е разрешен. Критерият може да бъде изразен техническо ограничение на ниво стратегия със стойност “dualStack”’=NO.

Критериите могат да бъдат разделени на две основни групи – технически и бизнес.

Техническите са наложени от околната среда. Те се асоциират със задължителни ограничения, наложени от текущата мрежа и поддържаните от нея услуги. Примери за такива в случая на преход от IPv4 към IPv6 са невъзможност за прилагане на един или друг механизъм за преход. Например, ако се използва транслация, това би могло да попречи на нормалното функциониране на услугите, свързани с предаването на глас или ако се използва двоен IP стек, това би изисквало ресурси, с които мрежата не разполага.

Бизнес ограниченията са наложени от различните заинтересовани лица. Например техническата дирекция би държала на стратегия с възможно по-ниски нива на риск. Маркетинг директорът на възможно по-кратко време за въвеждане на нови услуги, а финансовият - на балансиран бюджет и разширения на мрежата само и единствено на база на текущите приходи. Бизнес ограниченията могат да имат различен приоритет. Примери за подобни ситуации са често срещащи се - изискванията на финансовия и маркетинговия отдел имат по-висок приоритет от тези на техническата дирекция.

3.3.12.2 Алгоритъм

Алгоритъмът за избор на еволюционния път (Фигура 3-9)определя най-подходящата стратегия за еволюция на мрежата от текущото до желаното състояние. Алгоритъмът е демонстриран с MS Visio flowchart диаграма. Основните стъпки са:

Input – изпълнява ролята на точка за подаване на входящи параметри. Входящите параметри са Evolution criteria (еволюционни критерии) и формули за тяхното изчисление за всяка една стратегия и стратегии.

getNextStrategy - подава следващата стратегия от списъка със стратегии. Ако няма повече стратегии връща null.

NoMoreStrategies – проверка дали getNextStrategy не е върнал null. Ако резултатът от проверката е отрицателен, алгоритъмът продължава с предефинирания процес „Calculation of Evolution Criteria”. Ако проверката е положителна, алгоритъмът продължава с предефинирания процес „Determine the Evolution Path”.

Calculation of Evolution Criteria – предефинирания процес (Фигура 3-10) изчислява стойностите на еволюционните критерии на база на стойностите на технически и бизнес ограничения на стратегиите и изграждащите ги стъпки. Получава като входящи параметри еволюционни критерии, формули за тяхното изчисление и ограниченията валидни за дадената стратегия. Изходът на процеса връща резултат от изчислени еволюционни критерии за дадена стратегия.

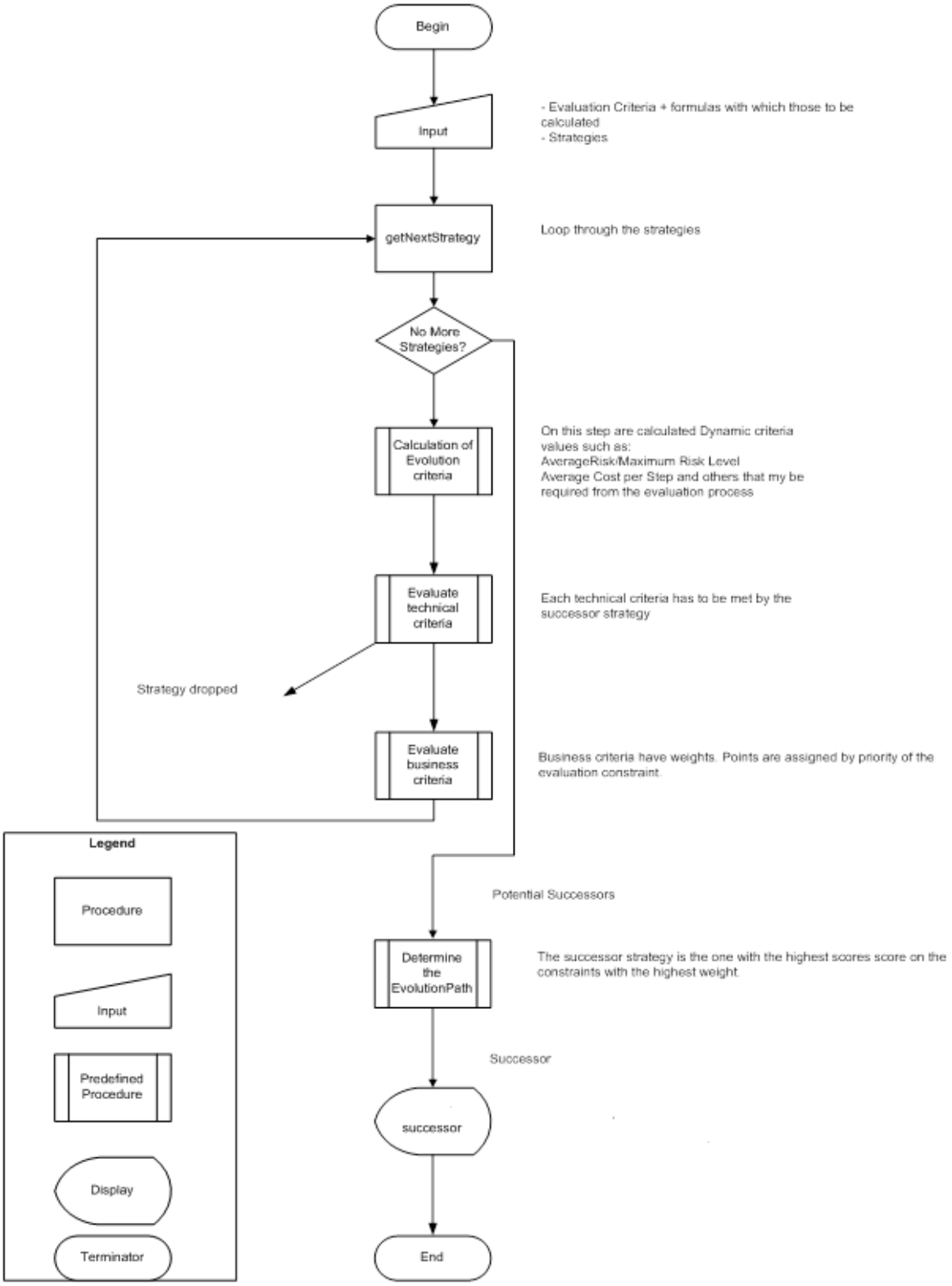
Evaluate technical criteria – предефинирания процес за оценка на техническите критерии (Фигура 3-11) проверява дали стратегията кандидат отговаря стриктно на зададените технически критерии. Ако това не е така, дадената стратегия отпада от процеса за избор на еволюционния път.

Evaluate business criteria - ако стратегията отговаря на техническите изисквания, се изследват стойностите на бизнес критериите (Фигура 3-12). Изискванията на бизнеса могат да бъдат с различна важност и е необходимо да бъдат степенувани. Това се постига чрез дефинирането на коефициент на тежест за всеки един критерий.

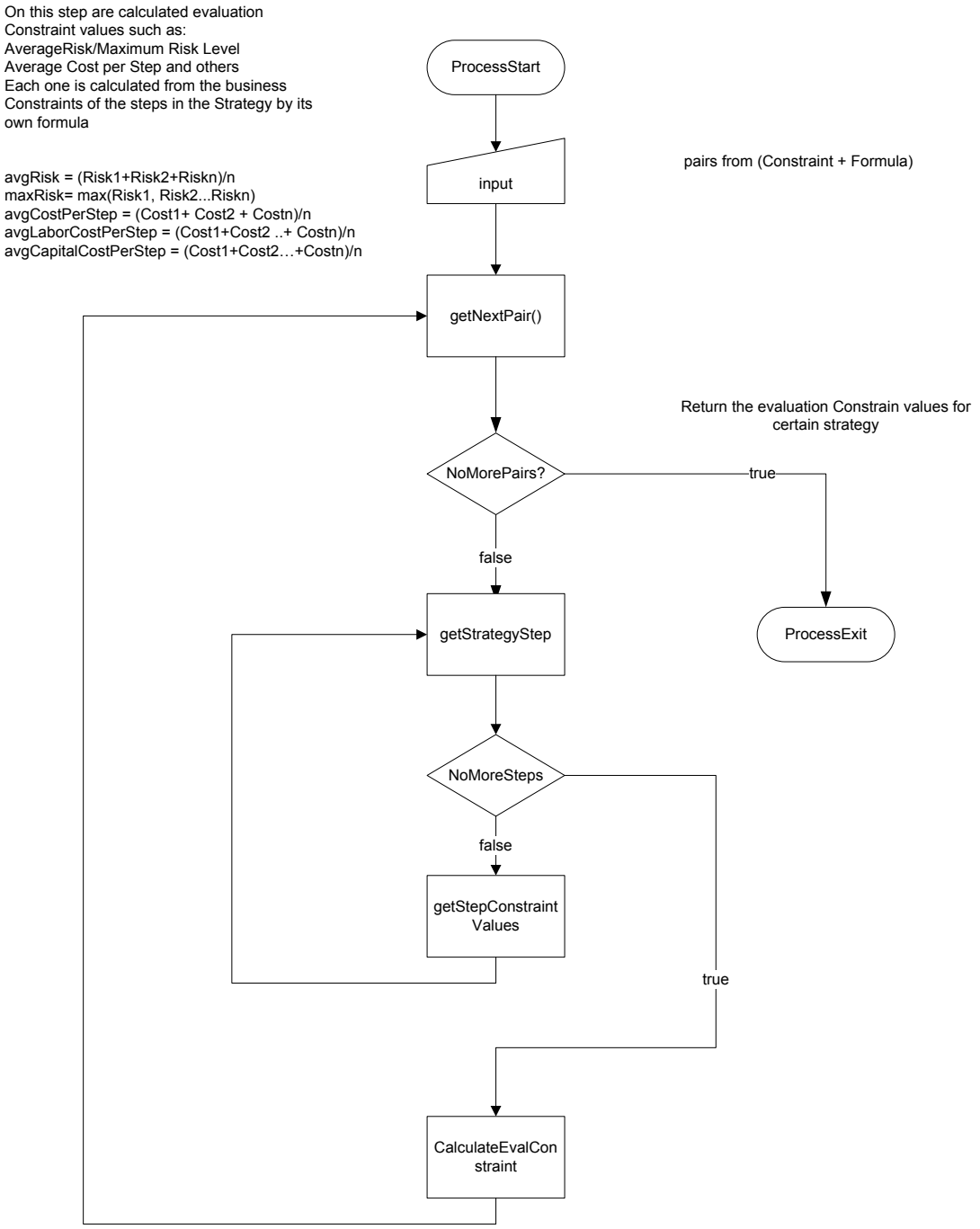
Determine the Evolution Path - след оценката на всяка една от стратегиите се избира стратегията победител “Successor” на базата на оценките направени в преходните стъпки (Фигура 3-13). Стратегията победител е тази, която изпълнява напълно техническите критерии за избор и най-добре отговаря на критериите, наложени от бизнеса. Това се постига чрез подредба на стратегиите според стойностите на коефициентите за тежест на еволюционните критерии. Ако критериите са цена, риск и време с тежести 3, 2, 1, то стратегията победител би била тази с резултат, при който общата тежест от всички критерии е максимална. Възможно, макар и малко вероятно в реална ситуации, е да има и две или повече стратегии, които да отговарят на еволюционните критерии. В този случай бива избрана стратегията, отговаряща на комбинацията от критерии с най-висока тежест. Ако въпреки това, стратегиите са с равен резултат, се избира стратегията с минимален брой стъпки.

Successor - алгоритъмът връща стратегията победител.

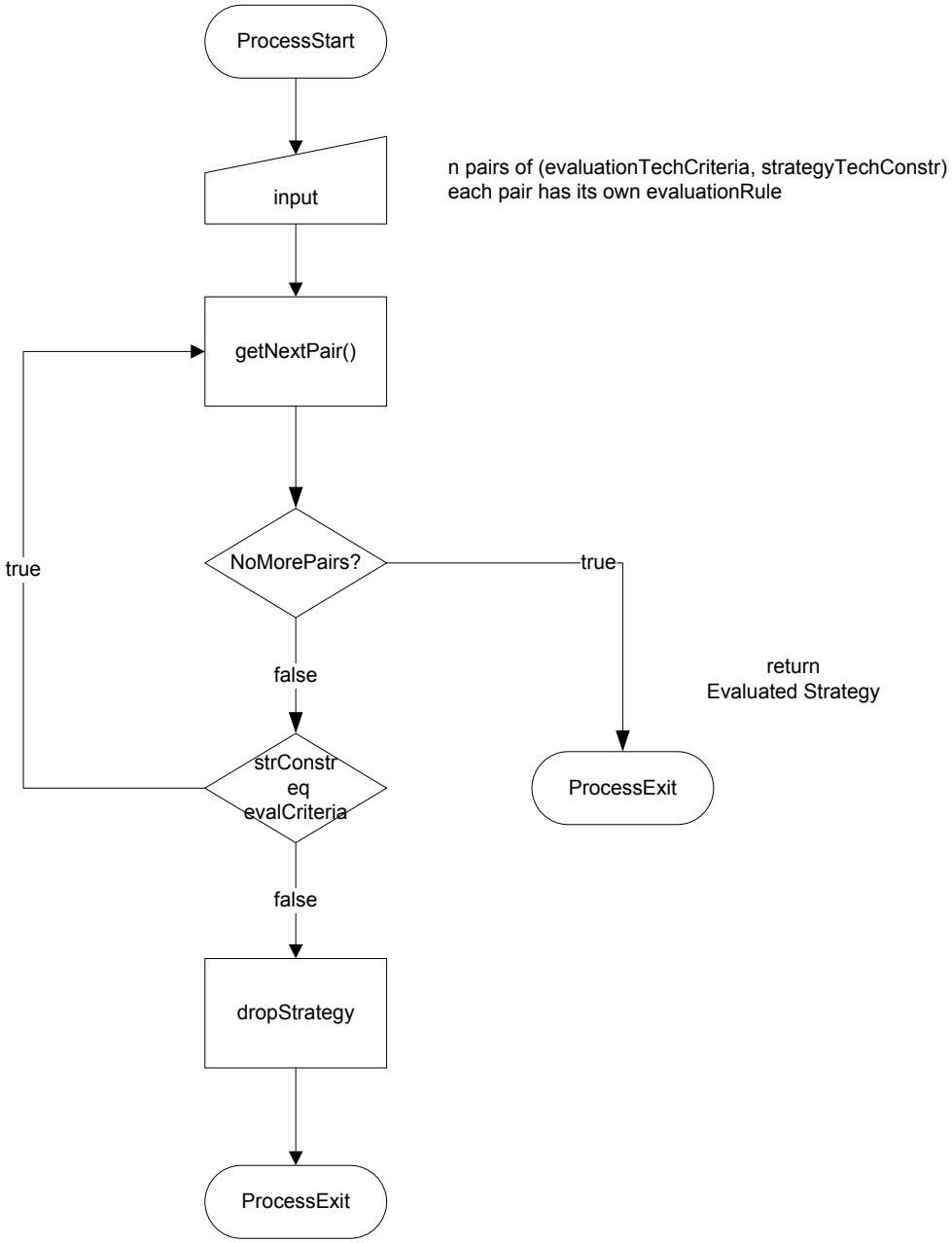
Фигура 3-9 Алгоритъм за избор на еволюционния път



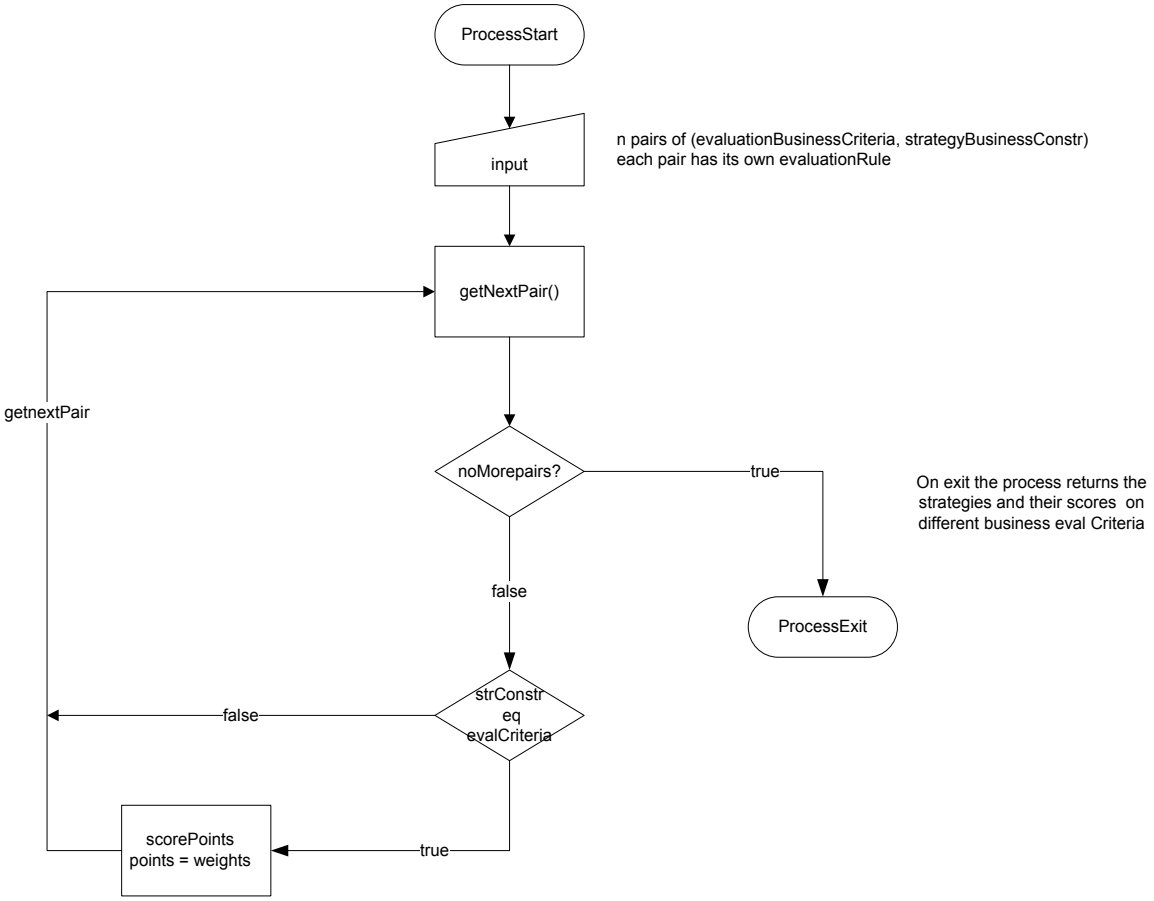
Фигура 3-10 Criteria Calculation (Изчисление на стойностите на критериите за оценка)



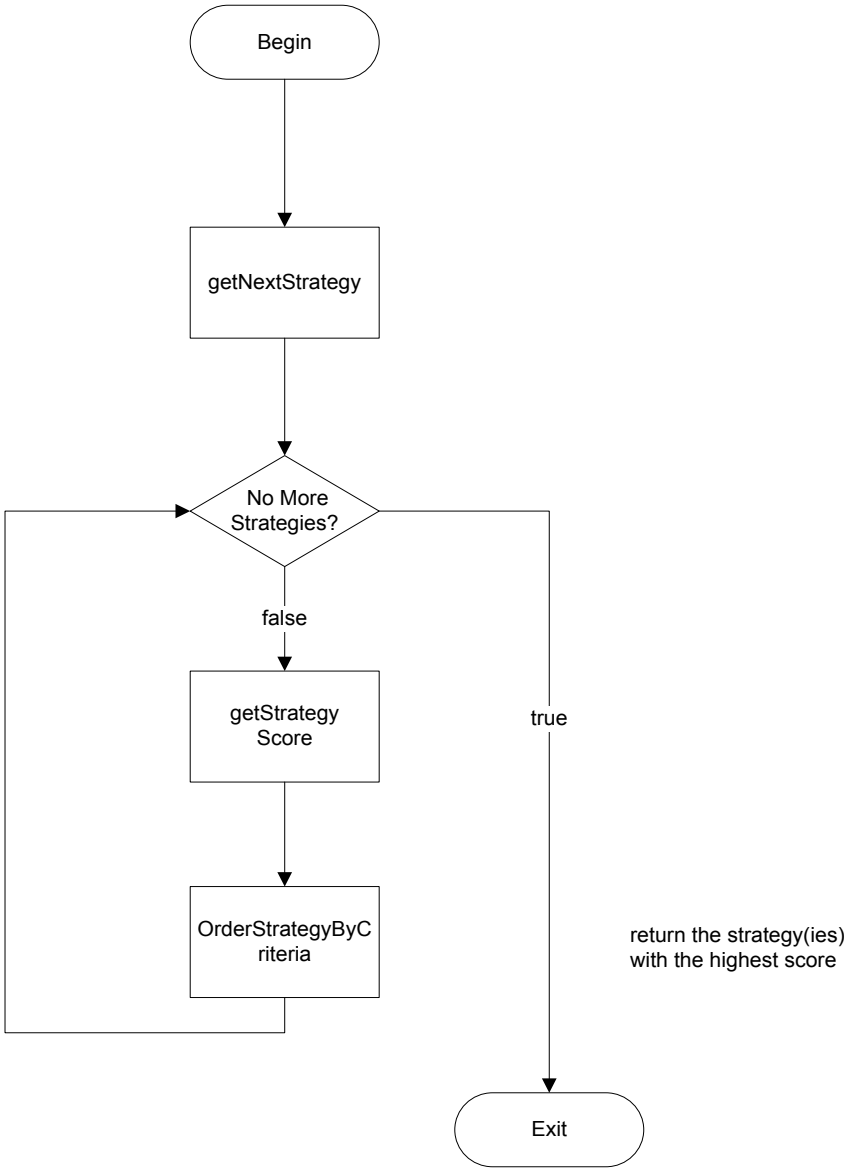
Фигура 3-11 Technical Criteria Evaluation (Оценка на техническите критерии)



Фигура 3-12 Business Criteria Evaluation (Оценка на бизнес критериите)



Фигура 3-13 Determine the evolution path (Определяне на еволюционния път)



3.3.13 Други алгоритми за решение на задачата

Задачата по вземането на решение коя е най-подходящата стратегия за преход от IPv4 към IPv6 спрямо контекста на даден мрежови оператор би могла да бъде решена не само чрез използването на описания от автора алгоритъм, но и по някой от множеството от многокритериални алгоритми за вземане на решения. Примери за подобни алгоритми са дадени в [84]. В настоящата дисертация е наблегнато на методологията и подхода до достигане на формални критерии, по които да бъде взето решение. Самото решение може

да бъде взето по много методи. Всяка една организация може да използва алгоритъма на автора, свой собствен метод или някоя от вече разработените и общоприети методологии за вземане на решение. Би било полезно да бъдат сравнен резултата от използването на алгоритъм за вземане на решение на базата на тегла, с резултата от алгоритъм базиран на методите за групово решение на задача (т.е метод при който решението се взема от множество заинтересовани лица). Решаването на подобни задачи и направата на подобно сравнение са сред бъдещите цели на автора.

3.4 Заключение

В глава 3 е разгледан подходът на автор за решение на проблем по трансформация на една мрежа от текущо към зададено желано състояние. Състоянията на мрежата са представени чрез формален графовиден модел. Преходът между двете състояния ще се извърши по път от множество други междинни състояния. Всяко едно от тях се достига чрез изпълнението на стъпка. Стъпката се състои от технически и бизнес ограничения, действие и ефект върху достигнатото състояние. Стъпките биват групирани в стратегии. Стратегиите могат да бъдат оценени по дадени еволюционни критерии. Критериите зависят от заинтересованите от трансформацията лица. Стратегията, която отговаря най-точно на критериите се избира за еволюционен път, по който ще бъде извършен прехода от текущо към желано състояние.

Глава 4: Приложение на подхода върху контекста на оператор X

4.1 Въведение

В глава 4 авторът ще демонстрира описания в глава 3 подход върху контекста на оператор X. За целта са моделирани първоначалното и желаното състояние на мрежата X, групи от стъпки за изпълнение на механизмите за преход от IPv4 към IPv6 и четири стратегии за цялостен преход между двете състояния. Стратегиите са подбрани на база на механизмите за преход описани в глава 1 и са базирани на [85]

Стратегиите са оценени по алгоритъма от глава 3 според критериите валидни за контекста на оператор X и е избрана най-подходящата стратегия за еволюция на мрежата.

4.2 Състояния

Първоначалното и желаното състояния на мрежата на X са моделирани на базата на проучване за преход от IPv4 към IPv6 извършено от автора в мрежата на реален мрежови оператор [10]. Детайлите на реалната мрежа са абстрактизирани и са оставени само най-съществените свойства на модела на устройствата.

4.2.1 Допускания

- Възлите, изпълняващи ролята на CE, поддържат IPv4, IPv6, NAT-PT и изграждане на тунели 6to4.

- P е обявено от производителя на техника в състояние “End-of-Live”, което не поддържа IPv6. Това устройство трябва да бъде заменено рано или късно с по-нов модел.

- Многофункционалното устройство DC изпълнява ролята на маршрутизатор, loadbalancer (възел, преразпределящ трафичните потоци към останалите машини в центъра за данни) и поддържа IPv6.

- По план мрежата трябва да бъде разширена с две допълнителни устройства на слоя за обединява на трафичните потоци – PE1, PE2.

- X използва OSPFv2 за IPv4 маршрутизиращ протокол и OSPFv3 за IPv6 маршрутизиращ протокол.

- Всички възли на мрежата без Srv и HG са представени чрез реални модели на производителя на мрежово оборудване Cisco Systems [86].

- Командите, използвани в действията също са според синтаксиса на операционните системи IOS (Internetwork Operating System) [87] и IOS-XR (IOS for High End Routers) [88].

4.2.2 Метаданни

Метаданните са характеристиките на елементите на графа - връзки и възли. В показаните по-долу модели са демонстрирани само метаданните, имащи пряко отношение към прехода от IPv4 към IPv6.

4.2.2.1 Първоначално състояние

Метаданните, с които се характеризират възлите в първоначалното състояние на мрежа са:

- deviceModel - Идентификатор на модела на устройството.
- ManagementIPv4Address - IPv4 Адрес, който се използва за комуникация с устройството от системите за управление на мрежата.
- Port - Идентификатор на порта. Едно устройство може да има повече от един вид порт.
- Ipv4Forwarding - Идентификатор за това дали версия 4 на IP протокола работи на даденото устройство.
- bgpLoacalAS - Идентификатор на BGP автономна системата.

Метаданните, с които се характеризират връзките в първоначалното състояние на мрежа са:

- ipv4Forwarding - Идентификатор за това дали версия 4 на IP протокола работи на дадения интерфейс.
- bgp4Forwarding - Идентификатор за това дали на съответната връзка се използва за IPv4 eBGP peering

- localIPv4Address - Идентификатор на локалния IPv4 адрес на входящия порт (sourceport).
- remoteIPv4Address - Идентификатор на IPv4 адреса на порта на устройството на другия край на връзката (targetport).
- ipv4Routing - Идентификатор на типа на IPv4 маршрутизиращия протокол
- mediaType - Идентификатор на типа на средата на връзката.

4.2.2.2 Желано състояние

- deviceModel - Идентификатор на модела на устройството.
- ManagementIPv6Address - IPv6 Адрес, който се използва за комуникация с устройството от системите за управление на мрежата.
- Port - Идентификатор на порта. Едно устройство може да има повече от един вид порт.
- ipv6Forwarding - Идентификатор за това дали версия 6 на IP протокола работи на даденото устройство
- bgpLoacalAS - Идентификатор на BGP автономна системата.

Метаданните, с които се характеризират връзките в желаната мрежа са:

- ipv6Forwarding - Версия за това дали версия 6 на IP протокола, работи на порта на даденото устройство.
- bgp6Forwarding - Идентификатор за това дали на съответната връзка се използва за IPv6 eBGP peering.
- localIPv6Address - Идентификатор на локалния IPv6 адрес на входящия порт (sourceport).
- remoteIPv6Address - Идентификатор на IPv6 адреса на порта на устройството на другия край на връзката (targetport).
- Ipv6Routing - Идентификатор на типа на IPv6 маршрутизиращия протокол
- mediaType - Идентификатор на типа на средата на връзката.

4.2.3 Списък със съкращения

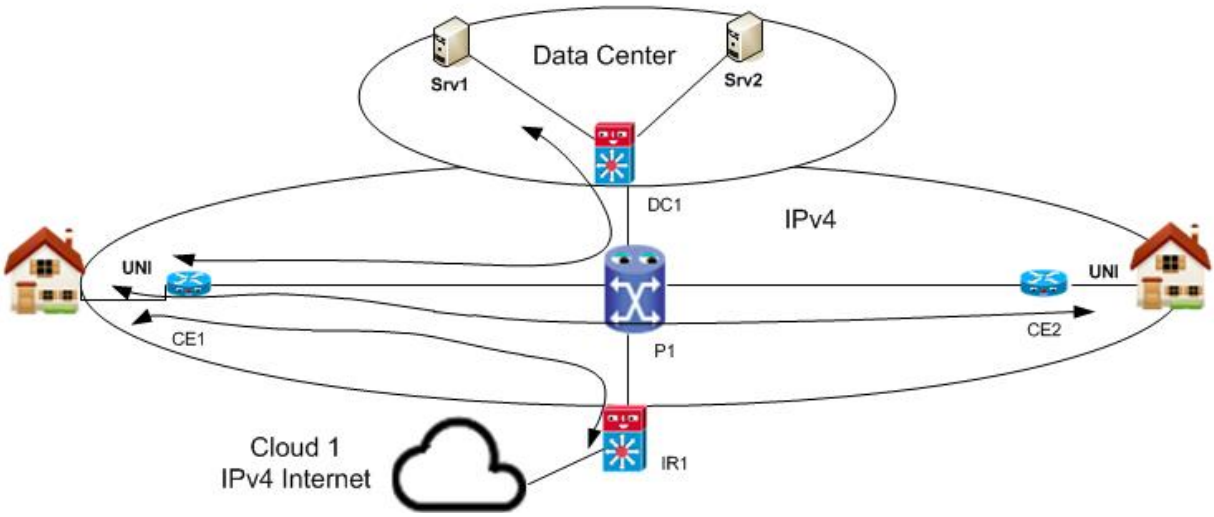
- CE (Customer Edge) – маршрутизатор, изграждащ и участващ в слоя за достъп в мрежата на оператора.

- P (Provider Core Router) – маршрутизатор, изграждащ опорната мрежа.
- DC (Data Center) – многофункционално устройство, свързващо центъра за данни с останалата мрежа.
- IR (Internet Router) - граничен маршрутизатор, свързващ оператора с IPv4 и IPv6 Интернет.
- Srv (Server) – сървър, разположен в центъра за данни. Може да е реална физическа машина или виртуална такава.
- HG (Home Gateway) – домашен шлюз. Свързва компонентите на услугата „Интелигентен дом" с останалата мрежа на оператора.
- PE (Provider Edge) – играе ролята на устройство, което обединява трафика от множеството устройства в слоя за достъп.
- UNI (User Network Interface) - интерфейсът между CE и HG.
- $Clould_1$- Сборен възел, играещ ролята на IPv4 Интернет.
- $Clould_2$ – Сборен възел, играещ ролята на IPv6 Интернет.

4.2.4 Модел на първоначално състояние на мрежата

Моделът на първоначалното състояние на мрежата се състои от един P, две CE, един DC и един IR маршрутизатор. В центъра за данни се намират сървъри Srv1 и Srv2.

Фигура 4-1 Първоначално състояние на мрежата



Според предложения подход, моделът на това състояние ще бъде изразен в Graphml формат (Фигура 4-2).

Фигура 4-2 Модел на състоянието на първоначалната мрежа

```
<graphml>
  <graph edgedefault="undirected">
    <key id="deviceModel" for="node" attr.name="deviceModel" attr.type="string"/>
    <key id="ManagementIPv4Address" for="node" attr.name="ManagementIPv4Address"
attr.type="string"/>
    <key id="port" for="node" attr.name="port" attr.type="string"/>
    <key id="ipv4Forwarding" for="node" attr.name="ipv6Forwarding" attr.type="string"/>
    <key id="bgp4Forwarding" for="edge" attr.name="bgp4Forwarding" attr.type="string"/>
    <key id="localIPv4Address" for="edge" attr.name="localIPv4Address" attr.type="string"/>
    <key id="remoteIPv4Address" for="edge" attr.name="remoteIPv4Address" attr.type="string"/>
    <key id="ipv4Routing" for="edge" attr.name="ipv4Routing" attr.type="string"/>
……<key id="bgpLoacalAS" for="node" attr.name=" bgpLoacalAS " attr.type="string"/>
……<key id="mediaType" for="edge" attr.name=" mediaType" attr.type="string"/>
    <node id="P1">
      <data key="port">Gig1/0</data>
      <data key="port">Gig0/0</data>
      <data key="port">Gig1/1</data>
      <data key="port">Gig2/1</data>
      <data key="deviceModel">cisco12810</data>
      <data key="ManagementIPv4Address">10.10.13.22</data>
      <data key="ipv4Forwarding">YES</data>
    </node>
```

```
 <node id="DC1">
   <data key="deviceModel">cisco7606</data>
   <data key="ManagementIPv4Address">10.10.14.2</data>
   <data key="ipv4Forwarding">YES</data>
   <data key="port">Gig1/0</data>
    <data key="port">Gig1/1</data>
    <data key="port">Gig1/2</data>
</node>
<node id="IR1">
   <data key="deviceModel">cisco7606</data>
   <data key="ManagementIPv4Address">10.10.15.2</data>
   <data key="ipv4Forwarding">YES</data>
   <data key="port">Gig1/0</data>
    <data key="port">Gig1/1</data>
    <data key="port">Gig1/2</data>
</node>
<node id="CE1">
   <data key="port">Fa1/0</data>
   <data key="port">Fa1/1</data>
   <data key="deviceModel">cisco2821</data>
   <data key="ManagementIPv4Address">10.10.10.1</data>
   <data key="ipv4Forwarding">YES</data>
</node>
<node id="CE2">
   <data key="port">Fa1/0</data>
   <data key="port">Fa1/1</data>
   <data key="deviceModel">cisco2821</data>
   <data key="ManagementIPv4Address">10.10.20.1</data>
   <data key="ipv4Forwarding">YES</data>
</node>
<node id="Srv1">
   <data key="port">Gig1/0</data>
   <data key="deviceModel">hpDL340L</data>
   <data key="ManagementIPv4Address">10.10.100.1</data>
   <data key="ipv4Forwarding">YES</data>
</node>
<node id="Srv2">
```

```
            <data key="port">Gig1/0</data>
            <data key="deviceModel">hpDL340L</data>
            <data key="ManagementIPv4Address">10.10.100.2</data>
            <data key="ipv4Forwarding">YES</data>
          </node>
          <node id="Cloud1">
      <!--NO port is defined here this is an external Host and even if there is a port we do not know it -->
      <!--Тук не е дефиниран физически порт. Дори и да има такъв ние не го знаем тъй като е на външен доставчик-->
            <data key="deviceModel">cloud</data>
            <data key="ManagementIPv4Address">87.115.300.2</data>
            <data key="ipv4Forwarding">YES</data>
          </node>
          <node id="House1">
      ……<!-- The model is left as a house. Infact this will be an end host device-->
      ……<!--Моделът е дефиниран като къща. Всъщност реално това е крайно устройство-->
             <data key="deviceModel">house</data>
             <data key="port">Fa1/0</data>
             <data key="ManagementIPv4Address">87.113.200.2</data>
            <data key="ipv4Forwarding">YES</data>
          </node>
          <node id="House2">
             <data key="deviceModel">house</data>
             <data key="port">Fa1/0</data>
             <data key="ManagementIPv4Address">87.113.300.2</data>
             <data key="ipv4Forwarding">YES</data>
          </node>
         <edge id=" CE1P1" source="CE1" target="P1" sourceport="Fa1/0" targetport="Gig1/1">
            <data key="localIPv4Address">10.10.13.21</data>
            <data key="remoteIPv4Address">10.10.13.22</data>
            <data key="ipv4Forwarding">YES</data>
            <data key="ipv4Routing">OSPF</data>
      ……..<data key="mediaType">FastEthernet</data>
          </edge>
           <edge id=" CE2P1" source="CE2" target="P1" sourceport="Fa1/0" targetport="Gig1/0">
            <data key="localIPv4Address">10.10.13.25</data>
            <data key="remoteIPv4Address">10.10.13.26</data>
```

```
            <data key="ipv4Forwarding">YES</data>
            <data key="ipv4Routing">OSPF</data>
        ……..<data key="mediaType">FastEthernet</data>
          </edge>
           <edge id="P1DC1" source="P1" target="DC1" sourceport="Gig1/1" targetport="Gig1/0">
            <data key="localIPv4Address">10.10.14.1</data>
            <data key="remoteIPv4Address">10.10.14.2</data>
            <data key="ipv4Forwarding">YES</data>
            <data key="ipv4Routing">OSPF</data>
        ……..<data key="mediaType">GigabitEthernet</data>
          </edge>
          <edge id="P1IR1" source="P1" target="IR1" sourceport="Gig2/1" targetport="Gig1/0">
            <data key="localIPv4Address">10.10.15.1</data>
            <data key="remoteIPv4Address">10.10.15.2</data>
            <data key="ipv4Forwarding">YES</data>
            <data key="ipv4Routing">STATIC</data>
        ……..<data key="mediaType">GigabitEthernet</data>
          </edge>
            <edge id="DC1Srv1" source="DC1" target="Srv1" sourceport="Gig1/2" targetport="Gig1/0">
            <data key="localIPv4Address">10.10.100.1</data>
            <data key="remoteIPv4Address">10.10.100.3</data>
            <data key="ipv4Forwarding">YES</data>
            <data key="ipv4Routing">STATIC</data>
        ……..<data key="mediaType">GigabitEthernet</data>
          </edge>
            <edge id="DC1Srv2" source="DC1" target="Srv2" sourceport="Gig1/3" targetport="Gig1/0">
            <data key="localIPv4Address">10.10.100.1</data>
            <data key="remoteIPv4Address">10.10.100.3</data>
            <data key="ipv4Forwarding">YES</data>
            <data key="ipv4Routing">STATIC</data>
        ……..<data key="mediaType">GigabitEthernet</data>
          </edge>
          <edge id="CE1House1" source="CE1" target="House1" sourceport="Fa1/1"
targetport="Fa1/0">
            <data key="localIPv4Address">87.113.200.1</data>
            <data key="remoteIPv4Address">87.113.200.2</data>
            <data key="ipv4Forwarding">YES</data>
```

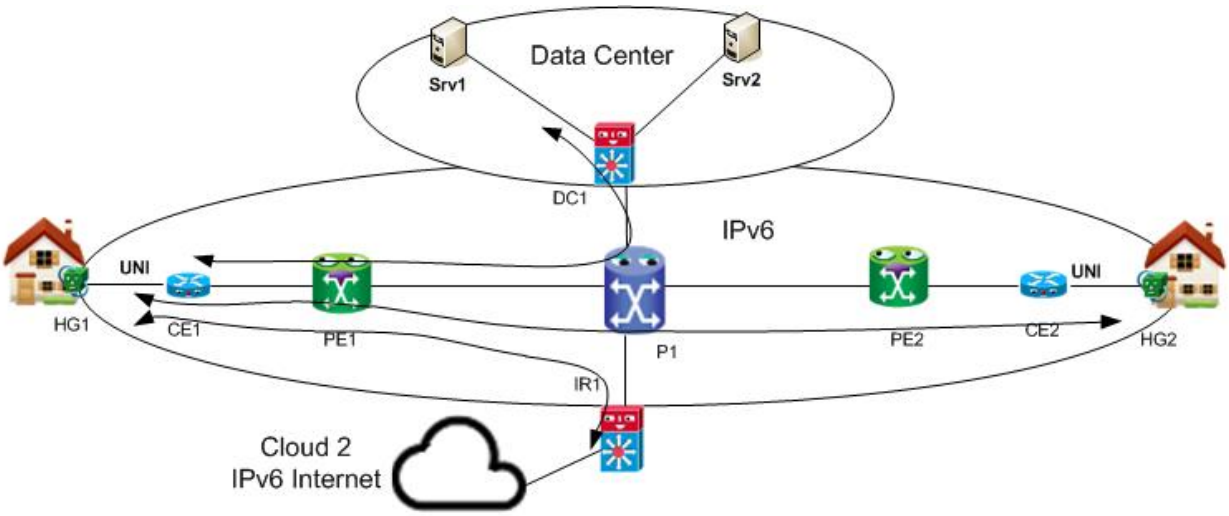
```
            <data key="ipv4Routing">STATIC</data>
        ……..<data key="mediaType">FastEthernet</data>
            </edge>
            <edge id="CE2House2" source="CE2" target="House2" sourceport="Fa1/1" targetport="Fa1/0">
              <data key="ipv4Forwarding">YES</data>
              <data key="localIPv4Address">87.113.201.1</data>
              <data key="remoteIPv4Address">87.113.201.2</data>
              <data key="ipv4Routing">STATIC</data>
        ……..<data key="mediaType">FastEthernet</data>
            </edge>
            <edge id="IR1Clould1" source="IR1" target="Cloud1" sourceport="Gig1/1">
              <data key="ipv4Forwarding">YES</data>
              <data key="localIPv4Address">87.114.201.1</data>
              <data key="remoteIPv4Address">87.114.201.2</data>
              <data key ="bgp4Forwarding">YES</data>
        ……..<data key="mediaType">GigabitEthernet</data>
            </edge>
          </graph>
        </graphml>
```

4.2.5 Желано състояние на мрежата

Моделът на желаното състояние на мрежата

На Фигура 4-3 е представена топология на желаната мрежа.

Фигура 4-3 Желана мрежа



Фигура 4-4 Модел на състоянието на желаната мрежа

```
<graphml xmlns:cmp="http://xsltsl.org/cmp">
  <graph edgedefault="undirected">
    <key id="deviceModel" for="node" attr.name="deviceModel" attr.type="string"/>
    <key id="ManagementIPv6Address" for="node" attr.name="ManagementIPv6Address"
attr.type="string"/>
    <key id="port" for="node" attr.name="port" attr.type="string"/>
    <key id="ipv6Forwarding" for="node" attr.name="ipv6Forwarding" attr.type="string"/>
    <key id="ipv6Forwarding" for="edge" attr.name="ipv6Forwarding" attr.type="string"/>
    <key id="bgp6Forwarding" for="edge" attr.name="bgp6Forwarding" attr.type="string"/>
    <key id="localIPv6Address" for="edge" attr.name="localIPv6Address" attr.type="string"/>
    <key id="remoteIPv6Address" for="edge" attr.name="remoteIPv6Address" attr.type="string"/>
    <key id="ipv6Routing" for="edge" attr.name="ipv6Routing" attr.type="string"/>
……<key id="bgpLoacalAS" for="node" attr.name=" bgpLoacalAS " attr.type="string"/>
……<key id="mediaType" for="edge" attr.name=" mediaType" attr.type="string"/>
    <node id="P1">
      <data key="port">Gig0/0</data>
      <data key="port">Gig1/0</data>
      <data key="port">Gig1/1</data>
      <data key="port">Gig2/1</data>
      <data key="deviceModel">CRS1</data>
      <data key="ManagementIPv6Address">FEC0:10:10:13::22</data>
      <data key="ipv6Forwarding">YES</data>
    </node>
     <node id="DC1">
```

```
    <data key="deviceModel">cisco7606</data>
    <data key="ManagementIPv6Address">FEC0:10:10:14::2</data>
    <data key="ipv6Forwarding">YES</data>
      <data key="port">Gig1/0</data>
     <data key="port">Gig1/1</data>
      <data key="port">Gig1/2</data>
</node>
<node id="IR1">
    <data key="deviceModel">cisco7606</data>
    <data key="ManagementIPv6Address">FEC0:10:10:15::2</data>
    <data key="ipv6Forwarding">YES</data>
    <data key="port">Gig1/0</data>
     <data key="port">Gig1/1</data>
     <data key="port">Gig1/2</data>
</node>
 <node id="PE1">
    <data key="deviceModel">cisco7401ASR</data>
    <data key="ManagementIPv6Address">FEC0:10:10:11::2</data>
    <data key="ipv6Forwarding">YES</data>
    <data key="port">Gig1/0</data>
     <data key="port">Gig1/1</data>
     <data key="port">Gig1/2</data>
</node>
 <node id="PE2">
    <data key="deviceModel">cisco7401ASR</data>
    <data key="ManagementIPv6Address">FEC0:10:10:12::2</data>
    <data key="ipv6Forwarding">YES</data>
    <data key="port">Gig1/0</data>
    <data key="port">Gig1/1</data>
    <data key="port">Gig1/2</data>
</node>
<node id="CE1">
     <data key="port">Fa1/0</data>
      <data key="port">Fa1/1</data>
    <data key="deviceModel">cisco2821</data>
    <data key="ManagementIPv6Address">FEC0:10:10:10::1</data>
    <data key="ipv6Forwarding">YES</data>
```

```
            </node>
            <node id="CE2">
               <data key="port">Fa1/0</data>
                <data key="port">Fa1/1</data>
              <data key="deviceModel">cisco2821</data>
              <data key="ManagementIPv6Address">FEC0:10:10:20::1</data>
              <data key="ipv6Forwarding">YES</data>
            </node>
            <node id="Srv1">
               <data key="port">Gig1/0</data>
              <data key="deviceModel">hpDL340L</data>
              <data key="ManagementIPv6Address">FEC0:10:10:100::1</data>
              <data key="ipv6Forwarding">YES</data>
            </node>
            <node id="Srv2">
               <data key="port">Gig1/0</data>
              <data key="deviceModel">hpDL340L</data>
              <data key="ManagementIPv6Address">FEC0:10:10:100::2</data>
              <data key="ipv6Forwarding">YES</data>
            </node>
            <node id="Cloud2">
              <data key="deviceModel">cloud</data>
              <data key="ipv6Forwarding">YES</data>
        <!--NO port is defined here this is an external Host and even if there is a port we do not know it -->
        <!--Тук не е дефиниран физически порт. Дори и да има такъв ние не го знаем тъй като е на външен доставчик-->
              <data key="deviceModel">cloud</data>
              <data key="ManagementIPv6Address">87:115:201::2</data>
              <data key="ipv6Forwarding">YES</data>
            </node>
            <node id="House1">
              <data key="deviceModel">house</data>
              <data key="port">Fa1/0</data>
              <data key="port">Fa1/1</data>
              <data key="ManagementIPv6Address">87:113:200::2</data>
              <data key="ipv6Forwarding">YES</data>
            </node>
```

```
        <node id="House2">
          <data key="deviceModel">house</data>
          <data key="port">Fa1/0</data>
          <data key="port">Fa1/1</data>
          <data key="ManagementIPv6Address">87:113:201::2</data>
          <data key="ipv6Forwarding">YES</data>
        </node>
        <node id="HG1">
          <data key="deviceModel">homeGateway</data>
          <data key="port">Fa1/0</data>
          <data key="ManagementIPv6Address">87:116:201::2</data>
          <data key="port">Fa1/1</data>
          <data key="ipv6Forwarding">YES</data>
        </node>
         <node id="HG2">
           <data key="deviceModel">homeGateway</data>
           <data key="port">Fa1/0</data>
           <data key="port">Fa1/1</data>
           <data key="ipv6Forwarding">YES</data>
          <data key="ManagementIPv6Address">87:116:202::2</data>
        </node>
       <edge id=" CE1PE1" source="CE1" target="PE1" sourceport="Fa1/0" targetport="Gig1/1">
          <data key="localIPv6Address">FEC0:10:10:13::21</data>
          <data key="remoteIPv6Address">FEC0:10:10:13::22</data>
          <data key="ipv6Forwarding">YES</data>
          <data key="ipv6Routing"> OSPFv3</data>
.....<data key="mediaType">FastEthernet</data>
        </edge>
         <edge id=" CE2PE2" source="CE2" target="PE2" sourceport="Fa1/0" targetport="Gig1/0">
          <data key="localIPv6Address">FEC0:10:10:13::25</data>
          <data key="remoteIPv6Address">FEC0:10:10:13::26</data>
          <data key="ipv6Forwarding">YES</data>
          <data key="ipv6Routing">OSPFv3</data>
.....<data key="mediaType">FastEthernet</data>
        </edge>
        <edge id=" P1PE1" source="P1" target="PE1" sourceport="Gig1/1" targetport="Gig0/0">
          <data key="localIPv6Address">FEC0:10:10:16::1</data>
```

```
            <data key="remoteIPv6Address">FEC0:10:10:16::2</data>
            <data key="ipv6Forwarding">YES</data>
            <data key="ipv6Routing">OSPFv3</data>
……..<data key="mediaType">GigabitEthernet</data>
        </edge>
         <edge id=" P1PE2" source="P1" target="PE2" sourceport="Gig1/1" targetport="Gig1/0">
            <data key="localIPv6Address">FEC0:10:10:17::1</data>
            <data key="remoteIPv6Address">FEC0:10:10:17::2</data>
            <data key="ipv6Forwarding">YES</data>
            <data key="ipv6Routing">OSPFv3</data>
……..<data key="mediaType">GigabitEthernet</data>
        </edge>
         <edge id=" P1DC1" source="P1" target="DC1" sourceport="Gig1/1" targetport="Gig1/0">
            <data key="localIPv6Address">FEC0:10:10:14::1</data>
            <data key="remoteIPv6Address">FEC0:10:10:14::2</data>
            <data key="ipv6Forwarding">YES</data>
            <data key="ipv6Routing">OSPFv3</data>
……..<data key="mediaType">GigabitEthernet</data>
        </edge>
        <edge id="P1IR1" source="P1" target="IR1" sourceport="Gig2/1" targetport="Gig1/0">
            <data key="localIPv6Address">FEC0:10:10:15::1</data>
            <data key="remoteIPv6Address">FEC0:10:10:15::2</data>
            <data key="ipv6Forwarding">YES</data>
            <data key="ipv6Routing">STATIC</data>
……..<data key="mediaType">GigabitEthernet</data>
        </edge>
         <edge id="DC1Srv1" source="DC1" target="Srv1" sourceport="Gig1/2" targetport="Gig1/0">
            <data key="localIPv6Address">FEC0:10:10:100::1</data>
            <data key="remoteIPv6Address">FEC0:10:10:100::3</data>
            <data key="ipv6Forwarding">YES</data>
            <data key="ipv6Routing">STATIC</data>
……..<data key="mediaType">GigabitEthernet</data>
        </edge>
        <edge id="DC1Srv2" source="DC1" target="Srv2" sourceport="Gig1/3" targetport="Gig1/0">
            <data key="localIPv6Address">FEC0:10:10:100::1</data>
            <data key="remoteIPv6Address">FEC0:10:10:100::3</data>
            <data key="ipv6Forwarding">YES</data>
```

```
            <data key="ipv6Routing">STATIC</data>
        ……..<data key="mediaType">GigabitEthernet</data>
          </edge>
          <edge id="CE1House1" source="CE1" target="House1" sourceport="Fa1/1"
targetport="Fa1/0">
            <data key="localIPv6Address">87:113:200::1</data>
            <data key="remoteIPv6Address">87:113:200::2</data>
            <data key="ipv6Forwarding">YES</data>
            <data key="ipv6Routing">STATIC</data>
        ……<data key="mediaType">FastEthernet</data>
          </edge>
          <edge id="CE2House2" source="CE2" target="House2" sourceport="Fa1/1"
targetport="Fa1/0">
            <data key="localIPv6Address">87:113:201::1</data>
            <data key="remoteIPv6Address">87:113:201::2</data>>
            <data key="ipv6Forwarding">YES</data>
            <data key="ipv6Routing">STATIC</data>
        ……<data key="mediaType">FastEthernet</data>
          </edge>
          <edge id="HG1House1" source="HG1" target="House1" sourceport="Fa1/1" targetport
="Fa1/1">
            <data key="localIPv6Address">87:116:201::2</data>
            <data key="remoteIPv6Address">87:116:201::1</data>
            <data key="ipv6Forwarding">YES</data>
            <data key="ipv6Routing">STATIC</data>
        ……<data key="mediaType">FastEthernet</data>
          </edge>
          <edge id="HG2House2" source="HG2" target="House2" sourceport="Fa1/1" " targetport
="Fa1/1">
            <data key="localIPv6Address">87:116:202::2</data>
            <data key="remoteIPv6Address">87:116:202::1</data>
            <data key="ipv6Forwarding">YES</data>
            <data key="ipv6Routing">STATIC</data>
        ……<data key="mediaType">FastEthernet</data>
          </edge>
          <edge id="IR1Clould1" source="IR1" target="Cloud1" sourceport="Gig1/1">
            <data key="localIPv6Address">87:115.201::1</data>
            <data key="remoteIPv6Address">87:115:201::2</data>
```

```
        <data key="ipv6Forwarding">YES</data>
        <data key ="bgp6Forwarding">YES</data>
……..<data key="mediaType">GigabitEthernet</data>
     </edge>
   </graph>
</graphml>
```

4.3 Еволюция на модела

Преходът от IPv4 към IPv6 се изразява в множество еволюционни промени на мрежовата инфраструктура. Промените могат да бъдат изразени формално чрез разлики в топологията на мрежата и в разлики между елементите и метаданните на графа, представящ крайното (желано) и първоначално състояние на мрежата.

4.3.1 Прилики и разлики в топологията на мрежата

Топологичните прилики между двете състояния са:

- Възли CE1, CE2, House1, House2, IR1, DC1, Srv1, Srv2, P1 съществуват и в двете топологии.
- Връзки CE1-House1, CE2-House2, P1-DC1, P1-IR1, DC1-Srv1, DC2-Srv2 съществуват и в двете състояния.

Топологичните разлики между желаното и първоначалното състояние са:

Възли:

- Появили са се четири нови възли HG1, HG2, PE1, PE2, Internet Cloud2;
- Изчезнал е възел Internet Cloud1.

Връзки:

- Появили са се нови връзки: HG1-House1, HG2-House2, CE1-PE1, CE2-PE2, P-PE1, P-PE2.
- Изчезнали са връзки: CE1-P1, CE2-P2, IR1-Cloud1;

4.3.2 Прилики и разлики в метаданните на графа, възлите и връзките

Свойства ManagementIPv4Address и ipv4Forwarding, съществуващи на всеки един възел в първоначалната топология са изчезнали от модела и са заменени с ManagementIPv6Address и ipv6Forwarding.

Свойство deviceModel на възел P1 се е променило от 12810 на CRS1.

Свойства localIPv4Address, remoteIPv4Address, ipv4Forwarding, ipv4Routing, съществуващи на всяка една връзка в първоначалната топология са изчезнали от модела и са заменени с localIPv6Address, remoteIPv6Address, ipv6Forwarding, ipv6Routing.

Появило се е ново свойството bgp6Forwarding на връзката между IR1 и InternetCloud2

Запазило се е свойството mediaType.

4.3.3 Анализ на приликите и разликите

Еволюцията на мрежата от IPv4 към IPv6 се е отразила на топологията на мрежата по следните начини:

- С внедряването на услугата Интелигентен Дом се появил нов микро слой за достъп в къщата на крайния клиент. Появили са се нови устройства и нови връзки.
- По-време на прехода опорната инфраструктура на мрежата се е разширила с нов слой от PE устройства. Това е пряк резултат от политиката на оператора за експанзия в нови зони, но и като мярка спрямо нарасналия потребителски трафик от новопоявилите се услуги (в случая Интелигентен дом).
- IPv6 е въведен на всеки един възел от мрежовата инфраструктура Това се е изразява в нова адресна схема и в промени на метаданните на всеки един възел и връзка. Промените се изразяват подмяна на версия 4 с версия 6 на IP протокола на всяко едно устройство и на всеки един негов интерфейс. В подмяна на IPv4 адреса с IPv6 такъв и в подмяна на маршрутизиращия протокол. В опорната мрежа OSPFv2 е заменен с OSPFv3. По останалите устройства се е запазил типът на маршрутизацията - статичен, но вече е статична IPv6 вместо статична IPv4 маршрутизация.

- Типът на средата на протокола под IP се е запазил непроменен.
- Интернет BGP peering се е променил и доставчикът вече няма свързаност към остарелия IPv4 Интернет.

Подобна еволюционна промяна не може да стане с „магическа пръчка“ тя трябва да се случи на малки стъпки групирани в определена стратегия и чрез прилагане на някой от механизмите за преход, описани в глава 1.

4.4 Стъпки

Преходите между отделните състояния ще се извършат на “стъпки”. Стъпките биват различни видове. Може да има стъпки, свързани с разширения на мрежата и добавяне на нови устройства и връзки. Друг тип стъпка е подмяна на съществуващо устройство или промяна на съществуваща връзка. Третият и най-често срещан вид стъпка е свързана с промяна на параметрите (метаданните) на съществуващи устройства и връзки. Всяка една стъпка, т.е. преход между две състояния, се състои от действие, ефект върху мрежата и може да бъде обвързана с определени технически и бизнес ограничения. Понякога ефекта от дадено действие може да се различава от предварително очаквания такъв. Ако това е така, се изпълнява “rollback” стъпка, съдържаща действие обратно на предишното или команда, с която дадено устройство да бъде върнато към последната му работеща конфигурация.

Стъпките са дефинирани според шаблона в подточка 4.3.1 и са разделени в няколко основни групи. Всяка една група съдържа стъпки с подобна функционалност, но дефинирани за устройства на различни слоеве от мрежата.

Синтаксисът, използван за дефиниране на действието, ефекта, техническите и бизнес ограниченията следва дефинициите и допусканията, направени в подточка 4.2.2.

4.4.1 Шаблон за дефиниране на стъпки

Всяка една от стъпките е структурирана в следния шаблон (Таблица: 4-1).

Таблица: 4-1 Шаблон за описване на стъпка

Номер: S - Уникален идентификационен номер на стъпката

Име: Име на стъпката
Контекст: Контекст, в който стъпката ще бъде приложена.
Технически ограничения: Техническите ограничения, които са задължителни за изпълнението на стъпката. Входящи параметри – входящите параметри, необходими за изпълнението на условията в проверката. Проверка – проверка спрямо текущото състояние.
Бизнес ограничения : Бизнес ограниченията, асоциирани с дадената стъпка. В най-общия случай това е цената, на която може да се изпълни стъпката, рискът при изпълнението на стъпката и времето, за което може да бъде осъществена дадената стъпка.
Действие: Входящи параметри - входящите параметри, необходими за изпълнение на действието. Действие – действието, с което се характеризира дадената стъпка. Често действието се изразява в изпълнение на шаблон с команди.
Ефект : Входящи параметри - входящите параметри, необходими за проверка на ефекта. Проверка – проверка, удостоверяваща дали даденото действие е донесло нужния ефект върху мрежата.

В настоящата дисертация са разгледани множество стъпки, свързани с различни механизми за преход от IPv4 към IPv6, а също и с фундаментални дейности като подмяна на мрежово оборудване и нова конфигурация на мрежово оборудване според контекста на оператор X. Описанието на стъпките е в отделен документ – Приложение 1.

Стъпките от Приложение 1 са разработени според контекста на оператор X, но могат да се използват със сравнително малки изменения за преход на коя да е мрежа на друг оператор от IPv4 към IPv6. Не случайно в стъпките не се споменават индексите на конкретните устройствата. Например, ако в модела на мрежата на X има устройство P1, то в стъпките е дадено просто P_i; ако в модела има PE1 и PE2, в стъпките и стратегиите се споменава само за PE и PE_i. По аналогия, ако става дума за DC1, IR1, CE1 и Srv1,2, то се използват DCi, IR_i, CE_i и Srv_i. Това е направено с цел да бъде демонстрирано, че стъпката или стратегията засяга всички устройства, намиращи се в дадения слой на мрежата или изпълняващи дадена роля в нея. Пример за изцяло дефинирана стъпка е демонстриран в Таблица: 4-2. В Приложение 1 са обобщени предложените стъпки за мигриране на мрежата от текущото към желаното състояние.

Таблица: 4-2 Пример за напълно описана стъпка

Номер: S13
Име: Enable NAT-PT (Добавяне на NAT-PT)
Контекст: Добавянето на механизъм за превод на адреси от IPv4 към IPv6 има приложение основно на слой CE. Той би играл важна роля при въвеждане в експлоатация на услугата „Интелигентен дом“. Стъпката има смисъл в стратегии, при които центърът за данни, опорната мрежа и Интернет са все още изцяло IPv4, а интелигентният дом е IPv6 базиран.
Технически ограничения: Входящи параметри: **$id**= ‘$CE_i$’, **$model**=’cisco2821’; Проверки: 1. Проверка дали IPv6 е конфигуриран на устройство CE_i: count(//graphml/graph/node[contains(@id,$id)]/data[@key='ipv6Forwarding' and . =’YES’]) = 1

2. Проверка дали IPv6 е конфигуриран на интерфейса, сочещ към клиента:

count(//graphml/graph/edge[contains(@source,$id)]/data[@key='ipv6Forwarding' and .='YES']) = 1

Бизнес ограничения :

Likelihood = Medium, IMPACT=Low =>Risk=LOW

Cost = $2280 - Цената включва разходи само за труд (57 човеко часа по $40 на час) и е определена на база на допускането, че IR устройствата поддържат NAT46/ NAT64 и няма допълнителни разходи, свързани с подмяна на устройства или обновяване на софтуера на съществуващите.

Preparation Time = 40h, Lab Testing Time = 16 h, Maintenance Window Time = 1h – стъпката не отнема много време, ако бъде изпълнена за едно устройство. Разбира се очакванията са това да не бъде така и тя да бъде изпълнена върху много устройства. В такъв случай времето за изпълнение в живата мрежа (Maintenance Window Time) следва да се умножи по броя на устройствата.

Действие:

Входящи параметри:

###vars: username=user, password=pass, $ipv6prefix=' 2001:DB8::/96', $ipv4prefix=' 11.11.11.11', $NNI='Fa1/0', $UNI='Fa1/1';

Шаблон:

```
### read_until('(login:|user:|Username:)',3)
$username
### read_until('(Password:|password:)',3)
$password
### start read_until('.*#',3)
set cli  screen-length 0
configure terminal
ipv6 nat v6v4 source $ipv6prefix $ipv4prefix
interface $UNI
ipv6 nat
interface $NNI
ipv6 nat
### stop read_until
exit
### exit
```

Ефект:

Ефектът се състои в две проверки спрямо преходното състояние. Първият ред проверява дали NAT-PT работи на ниво устройство, а вторият дали NAT-PT е конфигуриран на ниво интерфейс.

Входящи параметри:

$id= 'CE$_i$'

Проверка:

```
count(//graphml/graph/node[contains(@id,$id)]/data[@key='ipv6NatPt' and . ='YES']) = 1
count(//graphml/graph/edge[contains(@source,$id)]/data[@key='ipv6NatPt' and .='YES']) = 2
```

4.5 Стратегии за преход от IPv4 към IPv6 чрез преминаване през междинно състояние "Building Automation in Production"

Всяка една от изложените стратегии цели да постигне пълен преход от IPv4 към IPv4 чрез преминаване на междинно състояние „Building Automation in Production", съответстващо на осъществяването на услугата „Интелигентен дом" в мрежата на оператор X.

4.5.1 Допускания:

- Цената на хардуера за подмяната на P е $500000.
- Общата цена на хардуера за добавянето на един PE маршрутизатор е $160000.

4.5.2 Определяне на бизнес ограниченията на база на анкета

Един от основните проблеми, свързани с прехода от IPv4 към IPv6 е липсата на ясен механизъм за оценка на бизнес ограниченията, свързани с изпълнението на всяка една от стъпките в дадена стратегия. Проблемът донякъде се състои във факта, че за да бъде дадена достатъчно точно определение, трябва да е ясен контекста, в който ще бъде изпълнена дадена стратегия.

За определяне на бизнес критериите за всяка една от стратегиите, част от настоящата теза, бе направена анкета с експерти от различни представители на бизнеса, свързани с мрежовите технологии. Анкетата се състоеше данни за от:

- Контекста на оператор X;
- Диаграми на първоначалното и желаното състояние на мрежата;
- Описание и диаграми на стратегиите за преход.

Всеки един от анкетираните трябваше да попълни таблица с бизнес ограниченията за всяка една от стъпките на стратегиите. Бизнес ограниченията бяха дефинирани като риск, цена и време.

Цената бе изразена в щатски долари.

Рискът в числови стойности, а именно:

- Note = 0;
- Low = 1;
- Medium = 2;
- High = 3;
- Critical = 4.

Времето е разделено на подвремена за:

- подготовка (Time for Preparation);
- тестване в лаборатория (Time for lab testing);
- изпълнение на стъпката върху живата мрежа (Maintenance Window).

Сред анкетираните имаше хора на инженерни позиции в телеком оператори и доставчици на услуги, представители на системните интегратори и на производителите на оборудване. Сред инженерите по Телекомуникации, които попълниха анкетата са:

инж. Тодор Емануилов – Началник отдел „Service IP and Corporate Solutions" в Cosmo Bulgaria Mobile, България

инж. Сезен Анефи – Sr. IT Consultant в „Intracom Svyaz", Русия

инж. Георги Рибарски – Senior Expert Analysis and Design в „Telelink", България

eng. Jovica Djordjevic – Solution & Product Manager IP в „Huawei Technologies", Германия

инж.Николай Манолов – Princial Engineer в „Juniper Networks", Великобритания

инж. Злати Петров – Инженер по телекомуникации и Изпълнителен Директор SNT Bulgaria

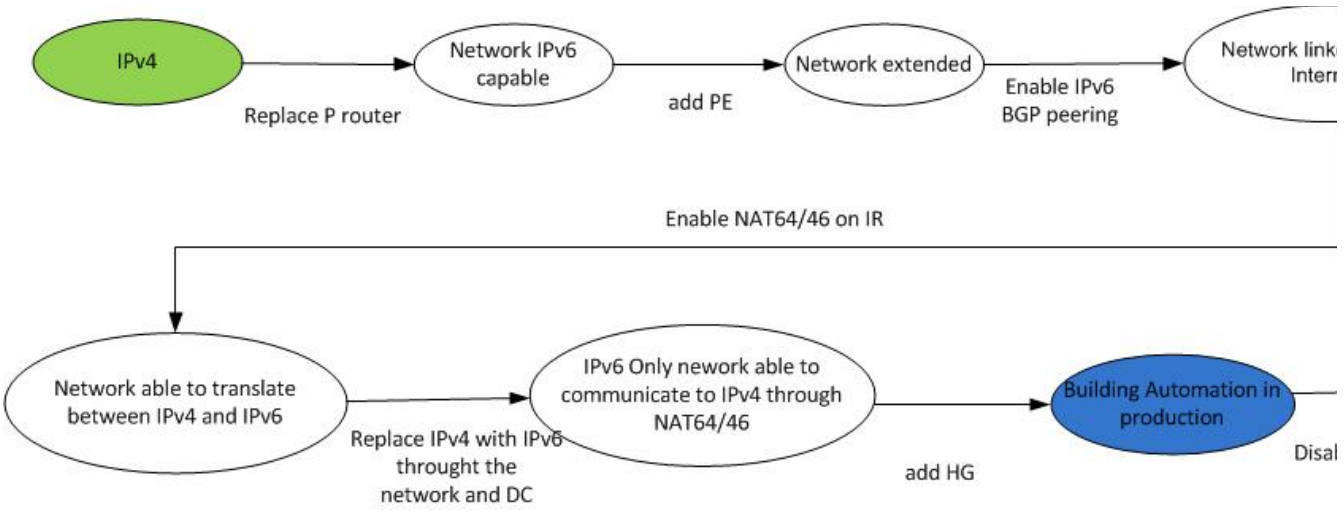
инж. Мартин Колев – Инженер по телекомуникации и Изпълнителен Директор TFN-T

Множество мнения бяха събрани и чрез публикуването на анкетата в социалната мрежа Linkedin и по специално в професионалните групи, свързани с прехода към IPv6.

4.5.3 Преход към IPv6 чрез stateless NAT64 и пълна подмяна на IPv4

Това е стратегия, позволяваща бърз, цялостен преход към IPv6 на цялата IPv4 инфраструктура (Фигура 4-5). В стратегията е заложена първоначална подмяна на оборудването, което не "говори" IPv6, цялостна трансформация на обновената инфраструктура към IPv6, въвеждане на "stateless" IP и ICMP NAT64 механизъм за комуникация със съществуващите IPv4 мрежи. След като мрежата бъде мигрирана и свързана със съществуващия IPv4 Интернет, следва въвеждането в експлоатация на услугата „Интелигентен дом". Последната стъпка на стратегията е изместена далеч в бъдещето и ще се състои в отстраняване на механизъма за превод на адреси към/ от IPv4 в момент, когато мрежите бъдат само IPv6. След изпълнението и на тази стъпка мрежата ще достигне желаното състояние „IPv6 - IPv6 Only".

Фигура 4-5 Fast-track strategy (стратегия без двоен IP стек)



4.5.3.1 Стъпки

- Replace P router – стъпката е свързана с големи капиталови разходи по подмяна на оборудване и води до състояние “Network IPv6 Capable”.

- Add PE router – стъпката, както и предишната е свързана с големи капиталови разходи. Тя трябва да бъде изпълнена на PE_1 и PE_2, и води до състояние “Network extended”.

- Enable IPv6 BGP peering – стъпката свързва мрежата на X с IPv6 Интернет пространството. Води до състояние “Network linked to IPv6 Internet”.

- Enable NAT64/ NAT46 on IR - стъпката позволява двупосочна комуникация между мрежата X и IPv4/ IPv6 Интернет. Води до състояние “Network able to translate between IPv4 и IPv6”.

- Replace IPv4 with IPv6 – стъпката трябва да бъде изпълнена на всяко едно устройство в мрежата на оператора (CE_1 и CE_2, PE_1 и PE_2, P, IR_1, DC_1, Srv_1 и Srv_2).Води до състояние “IPv6 Only nework able to communicate to IPv4 through NAT64/46”.

- Add HG – стъпката трябва да бъде изпълнена върху HG_1 и HG_2, и води до състояние “Building automation in Production”.

- Disable Stateless NAT64/NAT46 on IR – в случая стъпката води до състояние “IPv6 Only”.

4.5.3.2 Състояния

- IPv4 – начално състояние. Мрежата е изцяло IPv4 базирана.

- Network IPv6 capable – всеки един възел в мрежата е способен да поддържа IPv6. Мрежата на X достига това състояние след подмяната на устройството с име P_1, model=’cisco12810’ с такова с model=’CRS1’.

- Network extended – това междинно състояние се достига след добавянето на маршрутизатори PE_1 и PE_2 в мрежата на X. В модела на мрежата се добавят възли PE_1 и PE_2 и връзки между PE_1-CE_1 и PE_2- CE_2, PE_1-P, PE_2-P.

- Network linked to IPv6 Internet – състоянието отговаря на добавяне на нова връзка между IR_1 и $Cloud_2$. $Cloud_2$ е възел, който е обобщена презентация на IPv6 Internet.

- Network able to translate between IPv4 and IPv6 – в мрежата е добавена функционалност, позволяваща двупосочна комуникация между IPv4 и IPv6. Механизмът, използван за целта е NAT64/NAT46.

- IPv6 only network able to communicate to IPv4 through NAT64/46 –състоянието се достига след директен преход към IPv6 на всяко едно от съществуващите устройства.

- Building Automation in Production – това състояние е директно следствие от предходното. То се изразява в способността на мрежата да изпълни стъпките по добавянето на IPv6 HG.

- IPv6 Only – това е крайното желано състояние. То се достига след премахване на NAT64/46. Характеризира се с това, че всеки един възел има свойствата ipv4Forwarding със стойност NO и ipv6Forwarding със стойност YES.

4.5.3.3 Бизнес ограничения

Таблица: 4-3 Бизнес ограничения (Business Constraints) на стратегията за преход към IPv6 без двоен IP стек

Step	Target State	Risk	Cost	Time for		
				Preparation	Lab testing	Maintenance Window
Replace P	Network IPv6 Capable	3	523750	110	31	7
Add 2PE router	Network Extended	3	339645	82	32	10
Enable IPv6 BGP Peering	Enable IPv6 BGP	2	4345	34	18	4
Enable NAT46/64 on IR	Network able to translate between IPv6 and IPv4	2	6260	44	32	5
Replace IPv4 with IPv6	IPv6 only network able to communicate to IPv4 through NAT46/NAT64	3.2	13565	98	28	11
Add HG	Building Automation in production	1	2670	36	18	2
Disable Stateless NAT64/NAT46 on IR1	IPv6 only	2	3480	31	15	5

4.5.3.4 Потенциални критерии за избор

Стратегията би била подходящ избор, ако са налични следните ограничения:

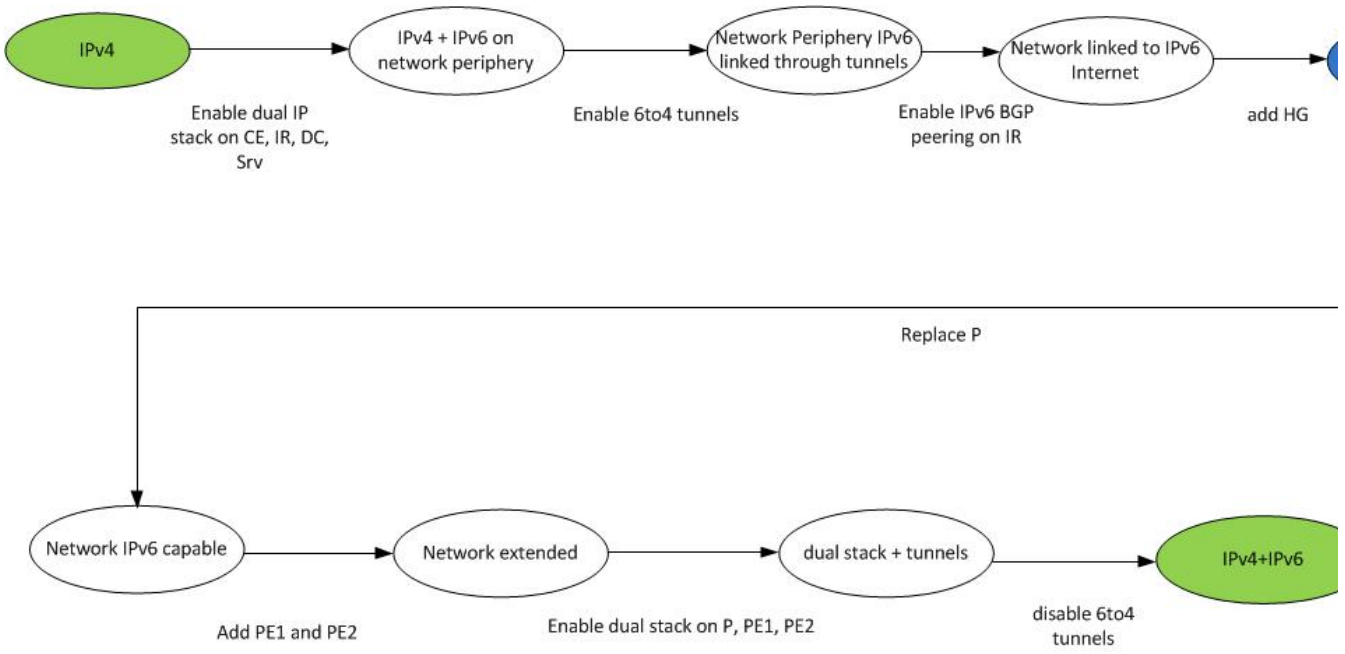
- Промените в опорната мрежа са позволени и възможни.
- Има първоначално наличен бюджет за разширения на мрежата и за подмяна на устройствата, които не поддържат IPv6.
- Няма точно фиксиран срок за въвеждането в експлоатация на новата услуга (т.е. “Интелигентният дом” може да почака).
- Допустими са сравнително високи нива на риск.
- Допустима е транслацията между IPv4 и IPv6.
- Двойният IP стек е нежелан механизъм за голяма част от възлите в мрежата (т.е. допустим е само там, където има транслация на адреси).
- Изграждането на тунели е нежелано.

4.5.4 Преход към IPv6 чрез изграждане на тунели и двоен IP стек

Стратегията позволява кратки пускови срокове за нови продукти, без поемане на излишни рискове и с минимални първоначални инвестиции. Стратегията е представена схематично на Фигура 4-6. В стратегията е заложена миграция към IPv6 чрез използване на двоен IP стек в слоя за достъп, центъра за данни и зоната за достъп до Интернет. Опорната мрежа остава непроменена и поддържа само IPv4. Поради тази причина се налага използването на 6to4 тунели между всички останали части на мрежата и през опорната мрежа. Налагането на подобна политика позволява сравнително ранно и "евтино" въвеждане в експлоатация на услугата "Интелигентен дом". Приходите от бъдещия продукт може да се използват за бъдещо разширяване на мрежата и за поетапна подмяна на оборудването в опорната мрежа. Веднъж подменено, опорната мрежа може да бъде трансформирана към IPv6. Това би позволило премахване на тунелите и миграция към пълен двоен стек. В последствие, при достигане на момент, в който IPv4 престане да се използва, двойният стек също ще трябва да бъде премахнат и да бъде достигнато желаното "IPv6 Only" състояние. Стратегията се характеризира с:

- диверсификация на риска;
- ранно въвеждане в експлоатация на потенциално печеливш продукт;
- отлагане на основните капиталови разходи, свързани с прехода към IPv6 за по-късен етап.

Фигура 4-6 Стратегия с изграждане на тунели и двоен IP стек



4.5.4.1 Стъпки

- Enable dual IP stack on CE, IR, DC, Srv - стъпката трябва да бъде изпълнена на CE_1 CE_2, IR_1, DC_1, Srv_1, Srv_2 и води до състояние “IPv4 + IPv6 on network periphery”.

- Enable 6to4 tunnels – стъпката трябва да бъде изпълнена между DC_1 и IR_1, CE_1 и IR_1, CE_2 и IR_1, CE_1 и DC_1 и между CE_2 и DC_1. Тя води до състояние “Network Periphery IPv6 linked through tunnels“.

- Enable IPv6 BGP peering on IR - стъпката трябва да се изпълни върху IR_1 и води до състояние “Network linked to IPv6 Internet”.

- Add HG – стъпката трябва да бъде изпълнена върху HG1 и HG2, и води до състояние “Building automation in Production”.

- Replace P router – стъпката е свързана с големи капиталови разходи по подмяна на оборудване и води до състояние „Network IPv6 Capable”.

- Add PE router – стъпката както и предишната е свързана с големи капиталови разходи. Тя трябва да бъде изпълнена на PE1 и PE2, и води до състояние “ Network extended“.

- Enable dual IP stack on P, PE_1 и PE_2.- стъпката конфигурира двоен IP стек на устройствата от опорната мрежа и води до състояние “dual ip stack + tunnels“.

- Disable 6to4 tunnels – стъпката трябва да бъде изпълнена между DC_1 и IR_1, CE_1 и IR_1, CE_2 и IR_1, CE_1 и DC_1 и между CE_2 и DC_1. Тя води до състояние на пълен двоен IP стек “IPv4+IPv6“.

- Disable dual IP stack – стъпката трябва да бъде изпълнена във варианта преход от двоен IP стек към IPv6 и води до крайното състояние “IPv6 only”.

4.5.4.2 Състояния

- IPv4 – начално състояние. Мрежата е изцяло IPv4 базирана.

- IPv4 + IPv6 on network periphery – състоянието се постига след въвеждане на двоен IP стек на устройствата извън опорната мрежа на оператора.

- Network Periphery IPv6 linked through tunnels – състоянието се достига след конфигурация на 6to4 тунели.

- Network linked to IPv6 Internet – състоянието отговаря на добавяне на нова връзка между IR_1 и $Cloud_2$. $Cloud_2$ е възел, който представлява IPv6 Internet.

- Building Automation in Production – състоянието се достига след добавянето на възли HG_1 и HG_2.

- Network IPv6 capable – всеки един възел в мрежата е способен да поддържа IPv6. Мрежата на X достига това състояние след подмяна на устройството с име P1 model='cisco12810' с такова с model='CRS1'.

- Network extended – това междинно състояние се достига след добавянето на маршрутизатори PE1 и PE2 в мрежата на X. В модела на мрежата се появяват възли PE1 и PE2, и връзки между PE1-CE1, PE2- CE2, PE1-P, PE2-P.

- Dual IP stack plus tunnels – състоянието отговаря на двоен IP стек в цялата мрежа и 6to4 тунели между CE, DC и IR.

- IPv4 + IPv6 (пълен двоен стек) – състоянието отговаря на двоен IP стек в цялата мрежа.

- IPv6 only - Това е крайното, желано състояние. То се достига след премахване на IPv4 от двойния IP стек.

4.5.4.3 Бизнес ограничения

Таблица: 4-4 Бизнес ограничения на стратегията за преход от IPv4 към IPv6 чрез изграждане на тунели и двоен IP стек

Step	Target State	Risk	Cost	Time for		
				Preparation	Lab testing	Maintenance Window
Enable dual IP stack on CE, IR, DC	IPv4 + IPv6 on network periphery	2	4980	40	25	8
Enable 6to4 tunnels on CE, IR, DC	Network Periphery IPv6 linked through tunnels	1	2131	27	20	5
Enable IPv6 BGP Peering	Enable IPv6 BGP	2	4345	34	18	4
Add HG	Building Automation in production	1	2088	21	11	2
Replace P	Network IPv6 Capable	3	523731	110	31	7
Add 2PE router	Network Extended	3	339296	82	32	10
Enable dual IP stack (P, PE)	Dual IP stack plus tunnels	2	2088	24	15	6
Disable 6to4 tunnels	IPv4 + IPv6 (пълен двоен стек)	1	1656	24	12	4
Disable dual IP stack	IPv6 only	2	1613	24	12	9

4.5.4.4 Потенциални критерии за избор

Стратегията за преход към IPv6 чрез двоен IP стек и изграждане на тунели ще бъде удачен избор в случай, че е налична комбинация от следните условия:

- Промените в опорната мрежа са силно нежелани и ако ги има, трябва да са с минимално ниво на риск.
- Няма първоначално наличен бюджет за разширения на мрежата и подмяна на оборудването, което не поддържа IPv6. Такъв е планиран в бъдеще.
- Има точно фиксиран и сравнително кратък срок за реализацията на новите продукти и услуги. Това означава, че услугата “Интелигентен дом” трябва да се случи възможно най-скоро и да е успешна, за да има финансови средства за бъдещи разширения на мрежата и за подмяна на старото оборудване.
- Високите нива на риск са недопустими или ако няма как, трябва да бъдат за много кратки интервали.
- Двойният IP стек е допустим механизъм.

- Механизмите за превод на адреси са несъвместими с услугите, предлагани от клиента.

- Допустимо е изграждането на тунели.

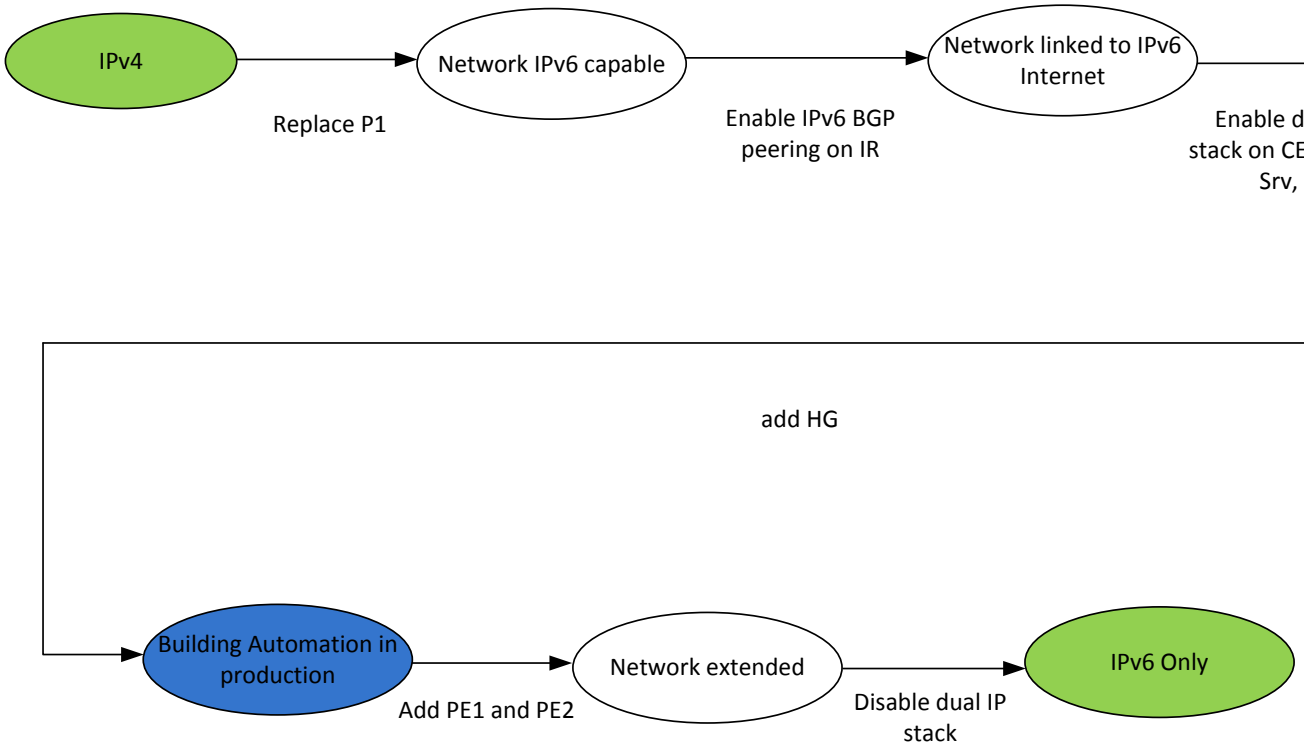
4.5.5 Преход към IPv6 чрез пълен двоен IP стек

Двоен IP стек на всяко едно мрежово устройство е сред класическите подходи за преход към IPv6 и е представен схематично на Фигура 4-7. Предложената в настоящата докторантура стратегия спрямо контекста на X, предполага изнесени напред във времето капиталови разходи за подмяна на оборудване и поетапно въвеждане на IPv6 на различните слоеве на мрежата. Миграцията към новия протокол започва от опорната мрежа, преминава през слоя за достъп, след това X се свързва към IPv6 Интернет пространството и накрая бива мигриран и центъра за данни. Последователността на тези стъпки не е от съществено значение. Важна в случая е стъпка 1, която гарантира, че мрежата ще бъде способна да работи с IPv6. След като мрежата е изцяло с двоен IP стек, вече може да бъде активирана услугата "Интелигентен дом". Новата услуга е базирана изцяло върху IPv6 и това е пример за стратегия, при която лесно биха се развивали нови IPv6 базирани услуги, а също и старите IPv4 такива. Последните няколко стъпки са разширение на слоя за обединение на трафичните потоци, забрана на IPv4 в някакъв бъдещ момент и достигане на желаното IPv6 състояние.

Стратегията предполага сравнително ниски нива на риск, но е свързана с потенциално големи първоначални разходи. Като най-голям недостатък може да се изтъкне факта, че операторът X ще трябва да поддържа двата протокола, както в равнината за данни, така и в контролната равнина в сравнително дълъг период от време. Това би увеличило сложността в мрежата и съответно разходите за поддържа, допълнително оборудване и персонал.

Стратегията не спестява IPv4 адресно пространство. Това би се случило, ако на база на двойния IP стек, X започне да развива изцяло IPv6 базирани услуги. Това едва ли би било възможно поради факта, че много от услугите ще трябва да комуникират с IPv4 Интернет, което предполага наличието или на допълнителен механизъм за превод на адреси, или използването на двоен стек с публични IPv4 адреси, т.е. разход на такива.

Фигура 4-7 Стратегия с пълен двоен стек



4.5.5.1 Стъпки

- Replace P router – стъпката е свързана с големи капиталови разходи по подмяна на оборудване и води до състояние “Network IPv6 Capable”.

- Enable IPv6 BGP peering on IR1 – стъпката позволява свързване на мрежата на X с IPv6 Интернет.

- Enable dual IP stack – стъпката трябва да бъде изпълнена на P, PE_1,PE_2, CE_1, CE_2, IR_1, DC_1 и Srv_1 и Srv_2 и води до състояние “IPv4+IPv6”.

- Add HG – стъпката трябва да бъде изпълнена за HG1 и HG2, и води до състояние “Building automation in Production”.

- Add PE router – стъпката, както и предишната, е свързана с големи капиталови разходи. Тя трябва да бъде изпълнена на PE1 и PE2, и води до състояние “Network extended”.

- Disable dual IP stack – стъпката трябва да бъде изпълнена във варианта преход от двоен IP стек към IPv6 за устройства P, PE_1,PE_2, CE_1, CE_2, IR_1, DC_1, Srv_1 и Srv_2 (таблици 1-2, 1-4, 1-6, 1-8, 1-10, 1-12 от Приложение 1).

4.5.5.2 Състояния

- IPv4 – начално състояние. Мрежата е изцяло IPv4 базирана.

- Network IPv6 capable – всеки един възел в мрежата е способен да поддържа IPv6. Мрежата на X достига това състояние след подмяна на устройството с име P1, model=’cisco12810’ с такова с model=’CRS1’.

- Network linked to IPv6 Internet – състоянието отговаря на добавяне на нова връзка между IR1 и Cloud2. Cloud2 е възел, който е обобщена презентация на IPv6 Internet.

- IPv4 + IPv6 (пълен двоен стек) – състоянието отговаря на двоен IP стек в цялата мрежа.

- Building Automation in Production – състоянието се достига след добавянето на възли HG1 и HG2.

- Network extended - състоянието се достига след добавянето на маршрутизатори PE1 и PE2 в мрежата на X. В модела на мрежата се появяват възли PE1 и PE2, и връзки между PE1-CE1 и PE2- CE2, PE1-P, PE2-P.

- IPv6 only – състоянието се достига чрез премахване на IPv4. То отговаря на желаното състояние.

4.5.5.3 Бизнес ограничения

Таблица: 4-5 Бизнес ограничения на стратегията за преход от IPv4 към IPv6 чрез двоен IP стек

Step	Target State	Risk	Cost	Time for		
				Preparation	Lab testing	Maintenance Window
Replace P	Network IPv6 Capable	3	523731	110	31	7
Enable dual IP stack (all devices)	IPv4 + IPv6	2	8030	53	27	15
Enable IPv6 BGP Peering	Enable IPv6 BGP	1	4345	34	18	4
Add HG	Building Automation in production	1	2088	21	11	2
Add dual-stack PE router	Network extended	3	339932	83	32	10
Disable dual IP stack	IPv6 only	2	3298	28	19	13

4.5.5.4 Критерии за избор

Стратегията за преход от IPv4 към IPv6 чрез двоен IP стек би била подходящ избор, ако са налични следните условия:

- промените в опорната мрежа са позволени и възможни;

- няма първоначално наличен бюджет за разширения на мрежата и за подмяна на устройствата, които не поддържат IPv6;

- няма точно фиксиран срок за въвеждането в експлоатация на новата услуга “Интелигентният дом”;

- не са допустими високите нива на риск;
- допустим е двойният IP стек;
- изграждането на тунели е нежелан механизъм;
- преводът на адреси е нежелан механизъм.

4.5.6 Преход към IPv6 чрез превод на адреси и двоен IP стек

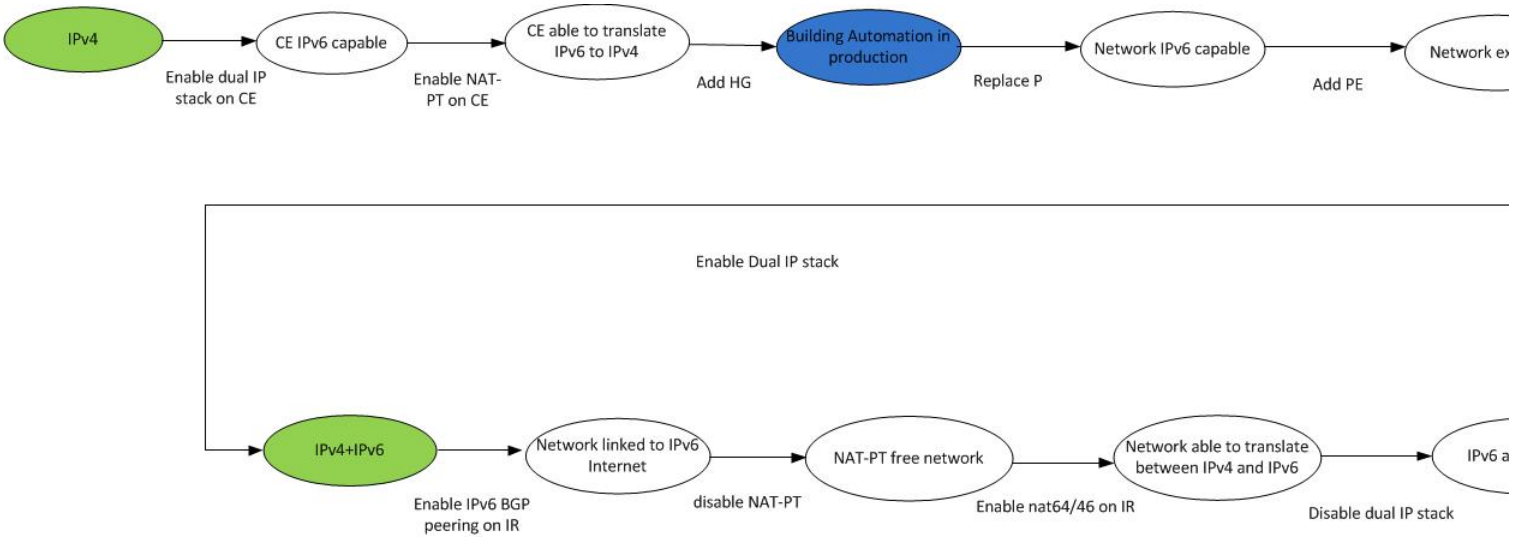
Преходът към IPv6 чрез прилагането на комбинация от механизми за превод на адреси и двоен IP стек е един от най-популярните методи за еволюция на IP мрежите. Стратегията, представена на Фигура 4-8 е съобразена с контекста на оператора X. Стратегията позволява бързо въвеждане на IPv6 в слоя за достъп чрез механизъма за превод на адреси NAT-PT и ранен старт на услугата „Интелигентен дом“. Ранният старт би дал техническо предимство на X пред останалите конкуренти. То би могло да се превърне в пазарно такова, ако услугата е успешна. Приходите от нея биха могли да се използват за инвестиции в ново оборудване (PE_1 и PE_2) и подмяна на съществуващо (P).

Следват стъпки по добавяне на двоен IP стек в мрежата на всички съществуващи възли (PE, P, DC_1 и IR_1). Достигането на междинно състояние Full Dual stack, позволява премахването на вече ненужните механизмите за превод на адреси в слоя за достъп. Последните стъпки са свързани с премахването на двойния IP стек и достигането на желаното състояние.

Стратегията предполага ниски нива на риск и използване на новите услуги за покриване на големи капиталови разходи, свързани с подмяна на оборудване и добавяне на ново такова. Недостатъкът на стратегията с двоен IP стек свързан с поддръжката на двата протокола е минимизиран чрез бързото въвеждането на NAT64 и премахването на единия стек (IPv4).

Стратегията първоначално не спестява IPv4 адресно пространство, но го прави в последствие чрез въвеждането на NAT64 и премахване IPv4 от двойния IP стек.

Фигура 4-8 Dual stack + NAT-PT+ NAT64



4.5.6.1 Стъпки

- Enable dual IP stack on CE - стъпката трябва да бъде изпълнена на CE_1 и CE_2. Тя води до състояние “CE IPv6 Capable”.

- Enable NAT-PT - стъпката трябва да бъде изпълнена на CE_1 и $CE_{2,}$, като двойният IP стек трябва да бъде конфигуриран на интерфейса към HG.

- Add HG – стъпката трябва да бъде изпълнена върху HG1 и HG2, и води до състояние “Building automation in Production”.

- Replace P router - стъпката е свързана с големи капиталови разходи по подмяна на оборудване и води до състояние „Network IPv6 Capable”.

- Add PE router – стъпката, както и предишната е свързана с големи капиталови разходи. Тя трябва да бъде изпълнена на PE1 и PE2, и води до състояние “Network extended“.

- Enable dual IP stack - стъпката трябва да бъде изпълнена на P, PE_1, PE_2, IR_1, DC_1, $Srv_{1,}$ и $Srv_{2,}$ и води до състояние “IPv4+IPv6”.

- Enable IPv6 BGP peering on IR1 - стъпката трябва да се изпълни върху IR1 и води до състояние “Network linked to IPv6 Internet”.

- Disable NAT-PT – стъпката трябва да се изпълни върху CE_1 и CE_2. Тя води до състояние “NAT-PT free network”.

- Enable NAT64/46 on IR_1 – стъпката се изпълнява върху IR_1 и дава възможност на мрежата за крос домейн комуникация с IPv4 и IPv6 интернет.

- Disable dual IP stack – стъпката трябва да бъде изпълнена във варианта преход от двоен IP стек към IPv6 (таблици 1-2, 1-4, 1-6, 1-8, 1-10, 1-12 от Приложение 1). Стъпката води до състояние “IPv4 free network”.

- Disable NAT64/46 – чрез стъпката се постига преход към желаното състояние “IPv6 only”.

4.5.6.2 Състояния

- IPv4 – начално състояние. Мрежата е изцяло IPv4 базирана.
- CE IPv6 capable – CE устройствата поддържат двоен IP стек.
- CE able to translate IPv6 to IPv4 - устройствата, които бъдат свързани след CE ще могат да комуникират по IPv6 с IPv4 домейна.
- Building automation in production – състоянието се достига след добавянето на възли HG1 и HG2.
- Network IPv6 capable - всеки един възел в мрежата е способен да поддържа IPv6. Мрежата на X достига това състояние след подмяна на устройството с име P1, model='cisco12810' с такова с model='CRS1'.
- Network extended - това междинно състояние се достига след добавянето на маршрутизатори PE1 и PE2 в мрежата на X. В модела на мрежата се появяват възли PE1 и PE2, и връзки между PE1-CE1, PE2- CE2, PE1-P, PE2-P.
- IPv4 + IPv6 (пълен двоен стек) – състоянието отговаря на двоен IP стек в цялата мрежа.
- Network linked to IPv6 Internet – състоянието отговаря на добавяне на нова връзка между IR1 и $Cloud_2$.
- NAT-PT free network – мрежа свободна от NAT-PT. Това е състоянието, което ще освободи IPv4 адресно пространство.
- Network able to translate between IPv4 and IPv6 – мрежа, способна да превежда адреси двупосочно между IPv4 и IPv6.
- IPv4 free network – състоянието се достига чрез премахване на IPv4 от всяко едно, от съществуващите устройства.
- IPv6 Only – желаното състояние.

4.5.6.3 Бизнес ограничения

Таблица: 4-6 Бизнес ограничения на стратегията за преход чрез NAT и двоен IP стек

Step	Target State	Risk	Cost	Time for		
				Preparation	Lab testing	Maintenance Window
Enable dual IP stack on CE	CE IPv6 Capable	1	1313	16	10	3
Enable NAT-PT on CE	CE able to translate IPv6 to IPv4	1	1633	16	10	3
Add HG	Building automation in production	1	3600	24	16	3
Replace P	Network IPv6 Capable	3	529233	128	33	10
Add 2PE router	Network Extended	3	348783	92	35	11
Enable IPv6 BGP Peering	Enable IPv6 BGP	2	9350	62	28	11
Enable dual IP stack	IPv4 + IPv6	1	6275	38	24	6
Disable NAT-PT on CE	NAT-PT free network	2	1200	16	8	4
Enable NAT46/64 on IR_1	Network able to translate between IPv4 and IPv6	2	2767	16	12	4
Disable dual IP stack	IPv6 only network able to communicate to IPv4 through NAT46/NAT64	2	1400	12	6	6
Disable NAT64/46	IPv6 only	2	1367	10	5	2

4.5.6.4 Потенциални Критерии за избор

Стратегията за преход от IPv4 към IPv6 чрез превод на адреси и двоен IP стек би била подходящ избор, ако са налични следните условия:

- Промените в опорната мрежа са силно нежелани и ако ги има, трябва да са с минимално ниво на риск.
- Промените в слоя за достъп са позволени.
- Няма първоначално наличен бюджет за подмяна на оборудването, което не поддържа IPv6. Такъв е планиран за даден бъдещ период.

- Разширенията на мрежата могат да бъдат отложени за даден бъдещ период.
- Новата услуга трябва да бъде въведена в експлоатация възможно най-бързо.
- Допустими са средните нива на риск.
- Допустим е двойният IP стек.
- Допустим е превода на адреси.
- Недопустимо е изграждането на тунели.

4.6 Оценка на стратегиите и избор на еволюционен път

Оценката на стратегиите се извършва според алгоритъма за избор на еволюционен път, описан подробно в глава 3. За целта първоначално ще бъдат дефинирани критерии за избор, съответстващи на интересите на различни заинтересовани лица от оператора X. След това ще се направят допускания на базата, на които ще може да се остойностят различните критерии.

Намирането на еволюционния път ще започне с предварителни изчисления на всеки един от критериите за избор за всяка една от стратегиите. След това ще се провери дали дадената стратегия отговаря на техническите критерии. Ако това е така, стратегиите ще бъдат подредени спрямо това как те отговарят и на бизнес критериите за оценка. Стратегията победител ще е тази, която отговаря на техническите критерии и е най-близка до изискванията на бизнеса.

4.6.1 Критерии за оценка и формули за тяхното изчисляване

Технически критерии:

- Преходът от IPv4 към IPv6 не може да се извърши чрез използване на механизми за изграждане на тунели.
- Допустимите механизми са двоен IP стек и превод на адреси

Бизнес критерии:

- Минималното време за прекъсване на мрежата до достигне на състояние “Building Automation In Production” .

- Ако две или повече стратегии имат едно и също време, то тогава се разглежда цената за достигане на състояние „Building automation in production”.
- Ако цените са еднакви, то се сравняват стойностите на максималния риск.
- Ако и те са еднакви, се взема под внимание крайната цена за достигане до желаното състояние (IPv6 only).
- Ако и цената на стратегиите съвпада се взема под внимание средното ниво на риска.

$$Ma\,\mathrm{int}\,enanceTimeToState > TotalCostToState > MaxRiskToState > TotalCost > AverageRiskToState$$

MaintenanceTimeToState – времето изразено във времеви интервали за поддръжка на мрежата за достигане до състояние Building Automation in Production. Времето за достигане на състояние “BuildingAutomation In Production” е най-важния критерии за избор:

$$MaintenanceTimeToState = \sum_{i=1}^{n} Ma\,\mathrm{int}\,enanceWindowTime_i, n \;=\; \mathrm{S}tate\ number$$

Total Cost To State – разходи за достигане до състояние “Building Automation in Production”:

$$TotalCostToState = \sum_{i=1}^{n} Cost_i,\ n \;=\; \mathrm{S}tate\ number$$

Max Risk to State – максимален риск до достигане на състояние “Building Automation in Production”:

$$MaxTRiskToState = \max_{i=1}^{n}(Risk_i), n = StateNumber$$

Average Risk To State - осреднен риск до достигане на състояние “Building Automation in Production”:

$$AvergageRiskToState = \frac{\sum_{i=1}^{n} Risk_i}{n}, n = StateNumber$$

Total Cost- разходите за изпълнение на цялата стратегия:

$$TotalCost = \sum_{i=1}^{n} Cost_i + RatePerHour * \sum_{i=1}^{n} TotalTime_i, n = FinalStateNumber$$

$$TotalTime = TimeFor\,\mathrm{Pr}eparation + TimeForLabTesting + Ma\,\mathrm{int}enanceWindowTime$$

Total Cost to State - разходите за изпълнение на стъпките до достигане на дадено състояние:

$$TotalCostToState = \sum_{i=1}^{n} Cost_i + RatePerHour * \sum_{i=1}^{n} TotalTime_i, n = StateNumber$$

$$TotalTime = TimeFor\,\mathrm{Pr}eparation + TimeForLabTesting + Ma\,\mathrm{int}enanceWindowTime$$

След изчислението на еволюционните бизнес критерии и оценката на техническите критерии се получава следната таблица. Стратегия номер 2 за преход чрез изграждане на тунели и двоен IP стек не отговаря на техническите критерии. Стратегии номер 1, 3 и 4 отговарят на техническите критерии и са потенциални кандидати за еволюционен път.

Таблица: 4-7 Пресмятане критериите за избор на стратегиите за трансформация на мрежата на X от IPv4 към IPv6

Strategy	Max Risk to State Building automation In production	Average Risk to IPv6 only	Total Cost To Building Automation in Production (USD)	Total Cost (USD)	Maintenance Time To State (days)
1	3.2	3.2	890235	893715	5.0
2	***1.8***	***2.8***	***13544***	***881928***	***2.275***
3	2.8	2.8	538194	881424	3.55
4	**1.4**	**2.8**	**6546**	**906920**	**1.3**

Последната фаза на алгоритъма е оценка на бизнес критериите. Според заданието MaintenanceTimeToState трябва да има минимална стойност и е с най-висок приоритет, а след него се нарежда TotalCostToState.

Сравнението показва, следното:

MaintenanceTimeToState(4)=1.3 < MaintenanceTimeToState(3)=3.55 <MaintenanceTImeToState(1) = 5.0
TotalCostToState(4)= **6546**< TotalCostToState(3)= 538194< TotalCostToState(1) = 890235

Следователно стратегия 4 (Преход чрез превод на адреси и двоен IP стек) отговаря най -точно на критериите за избор, наложени от заинтересованите лица на X и е стратегията, избрана за еволюционен път в мрежата на оператор X.

4.7 Заключение

В глава четири е приложен подходът на автора върху контекста на оператора X. Дефинирани са начално и желано състояние на мрежата на X. Началното състояние се състои от устройства, изцяло работещи с IPv4, а крайното - от устройства, изцяло работещи на IPv6. Преходът между двете състояние може да се извърши на множество стъпки. За всяка една стъпка са определени технически и бизнес ограничения, действие и ефект. Стъпките са групирани в стратегии. Всяка една от стратегиите е подходяща при определени изисквания от страна на заинтересованите лица и е потенциален кандидат за еволюционен път. Авторът е предложил четири стратегии за преход от IPv4 към IPv6.

Стратегията за преход чрез използването на механизми за превод на адреси и без двоен IP стек гарантира бързо достигане на желаното състояние (IPv6 only). Цената, на която това става, е високия риск и високите капиталови разходи, свързани с подмяна на оборудване в началото на стратегията. Важното междинно състояние “Building Automation in Production” се достига сравнително трудно и с цената на множество интервали за прекъсване на мрежата.

Стратегията за преход чрез двоен IP стек и изграждане на тунели позволява въвеждане в експлоатация на услугата „Интелигентен Дом“ на сравнително ниска цена и с умерена обща продължителност на периодите за поддръжка в мрежата. Като съществен неин недостатък се счита използването на тунели и поради тази причина тя е единствената стратегия, която не отговаря на техническите критерии за оценка.

Стратегията за преход чрез пълен двоен IP стек предлага олекотен и унифициран преход към IPv6. Цената, на която това е възможно, се определя от подмяната на оборудването, което според първоначалните допускания не поддържа IPv6. Тази подмяна, както и въвеждането в експлоатация на двойния IP стек, е свързана с продължителни прекъсвания на мрежата, които също не са толерирани от заинтересованите лица в X.

За еволюционен път е избрана стратегията за преход чрез двоен IP стек и механизми за превод на адреси. Тази стратегия използва NAT-PT за достигане на състояние “Building Automation in Production” и съответно успява да достави услугата „Интелигентен дом“ на много по-ниска цена от останалите стратегии. Времевите интервали, в които ще има евентуално прекъсване на услугите в мрежата също са минимални. Недостатък на стратегията е, че тя е най-скъпа от всичките четири варианта и е съставена от най-много стъпки. Това е така, поради факта, че механизмите, използвани за достигане на „Building Automation In Production“ се оказват неефективни с течение на времето и трябва да бъдат заменени с по-оптимални такива.

В глава четири е демонстриран подходът на автора за избор на еволюционен път на мрежа от IPv4 към IPv6. В следващата глава от дисертация е представен прототип на софтуер за автоматизиране на процеса по следене на състоянието на мрежата и за прилагане на стъпките от стратегията, избрана за еволюционен път.

Глава 5: Прототип на система за трансформация на мрежи от IPv4 към IPv6

5.1 Въведение

Една от основните цели на настоящата докторантура е създаването на софтуерен прототип, който да подпомага мрежовите инженери и архитекти в процеса на преход на дадена мрежа от едно състояние към друго и в частност от IPv4 към IPv6.

Основната цел на прототипа е да разкрива автоматично настоящето състояние на мрежата и да го променя, изпълнявайки действията от стъпките в стратегията, която е избрана за еволюционен път. Прототипът трябва да дава възможност на мрежовите архитекти и инженери да имат достоверна информация за топологията на мрежата на различни нива от OSI модела. Също така трябва да предоставя информация за видовете интерфейси между устройствата и за самите устройства. Друго основно изискване към прототипа е да предоставя възможност за визуализация на разликите между две състояния на ниво топология.

Архитектурен стил избран на база на изискванията и следван от авторът при създаването на прототипа на системата за трансформация на услуги е MVC (Model-Vew-Controller), представен в [89]. Изборът на този архитектурен стил позволява отделяне на дизайнерските решения свързани с модела (моделите) на мрежата от тези свързани с контролера и на визуализацията

Основното приложение на прототипа е в автоматизиране на прехода от IPv4 към IPv6, но той може бъде използван за в бъдеще и за редица други приложения. Примери за подобни могат да са масова подмяна на оборудване, въвеждане на нови услуги, разширения, реорганизации и подмяна на параметрите на съществуващи услуги в реални мрежови инфраструктури.

5.2 Анализ на съществуващите алгоритми за разкриване на мрежи

R. Siamwalla et al. [90] и Yuri Breitbart et al. [91] са направили добро проучване на методите за разкриване на мрежова топология на базата на ping, traceroute, SNMP, DNS и ARP. Като основен недостатък на тяхното проучване може да се посочи фокуса върху

топологията на мрежата на ниво 3 и липсата на механизми за откриване на топологията на ниво 2 от OSI модела. Lowekamp et al. [92] и Pandey et al. [93] са разработили механизъм за разкриване на хетерогенни среди, независимо от вида на мрежата, но техния механизъм изисква ICMP spoofing, за да разкрие препредаващата таблица на устройствата. ICMP spoofing се счита за техника, носеща сериозен риск за сигурността на мрежата и съответно е забранена в почти всяка една корпоративна мрежова среда. Това прави предложения от тях алгоритъм почти неизползваем.

Алгоритмите, разработени от тези автори, имат и друг съществен недостатък - разкриването на мрежата отнема значителен период от време. Една от основните причини за това е липсата на условия, които да ограничават процеса по разкриването на мрежата на базата на предварително зададени критерии. Реалната практика показва, че много рядко практическата задача е разкриване на цяла мрежа. В повечето случай е необходимо да бъде разкрит отделен регион или много по-малка част от мрежовата инфраструктура.

Не на последно място може да се спомене и факта, че нито един от авторите не предлага средства за динамично попълване на модел на мрежата на база на разкритата топология. Приложението на алгоритъма за разкриване на мрежата за динамично попълване на модел от данни, характеризиращи текущото състояние на мрежата липсва при всички.

Липсата на концепцията за модел оказва влияние и на средствата използвани за визуализация на мрежовата топология. Нито един от алгоритмите не предлага визуализация на топологията базата на дадени свойства на разкритите възли и връзки. Следствието от това е липсата на възможност за генериране на топология на мрежата на базата на комбинация от критерии – например топология на IPv6 или IPv4 мрежа на базата на протокол от слой 2 (CDP) и два протокола от слой 3 (OSPF, ISIS).

5.3 Алгоритъм за разкриване на настоящето състояние на мрежата

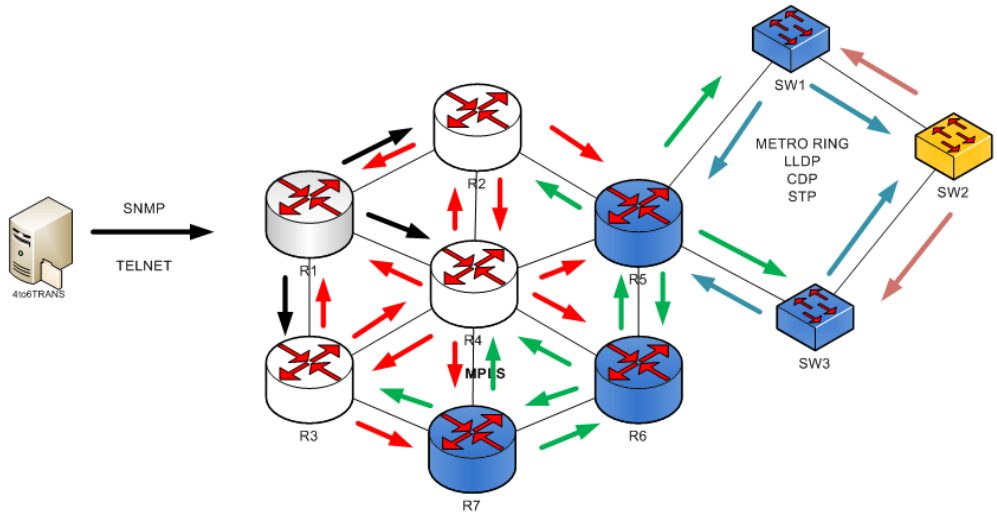
Алгоритмът за разкриване на топологията има за цел да открие всичките устройства и връзките между тях в мрежовата инфраструктура на доставчика на услуги. За да бъде възможно това, трябва да са изпълнени следните предварителни условия:

- Необходимо е ръчно да бъдат въведени IP адреса на първоначалното устройство и SNMP Community String, чрез който прототипът да може да комуникира с устройството.
- Всички устройства в дадената мрежа трябва да са с една и съща SNMP community настройка.
- Устройства трябва да имат поне един IP адрес.
- Устройствата трябва да имат пълна свързаност към IP възела, на който се изпълнява прототипа.
- Трябва да бъдат настроени stop-criteria (критериите за ограничение на процеса по разкриване на мрежата). Примери за подобни критерии са разкриване само на първото устройство или да бъдат разкрити само устройствата от даден слой на мрежата. Ако не бъдат настроени подобни критерии, алгоритъмът разкрива цялата мрежа.
- Устройствата, които бъдат разкрити чрез техните съседи и не отговарят на SNMP заявките, ще бъдат маркирани в топологията със следното наименование „Unknown-DeviceIPAddress”.

Процесът по разкриване на мрежата се изпълнява върху първоначално разкритото устройство. След това се изпълнява върху съседните устройства и след това на техните съседни устройства и така докато не бъде разкрита цялата мрежа или докато не бъдат изпълнени критериите за ограничение на процеса. Алгоритмът има вграден механизъм, който предпазва от повторно изпълнение върху едно и също устройство.

На Фигура 5-1 и Фигура 5-2 е демонстрирано как протича процесът по разкриване на мрежова топология.

Фигура 5-1 Протичане на процеса по разкриване на устройства

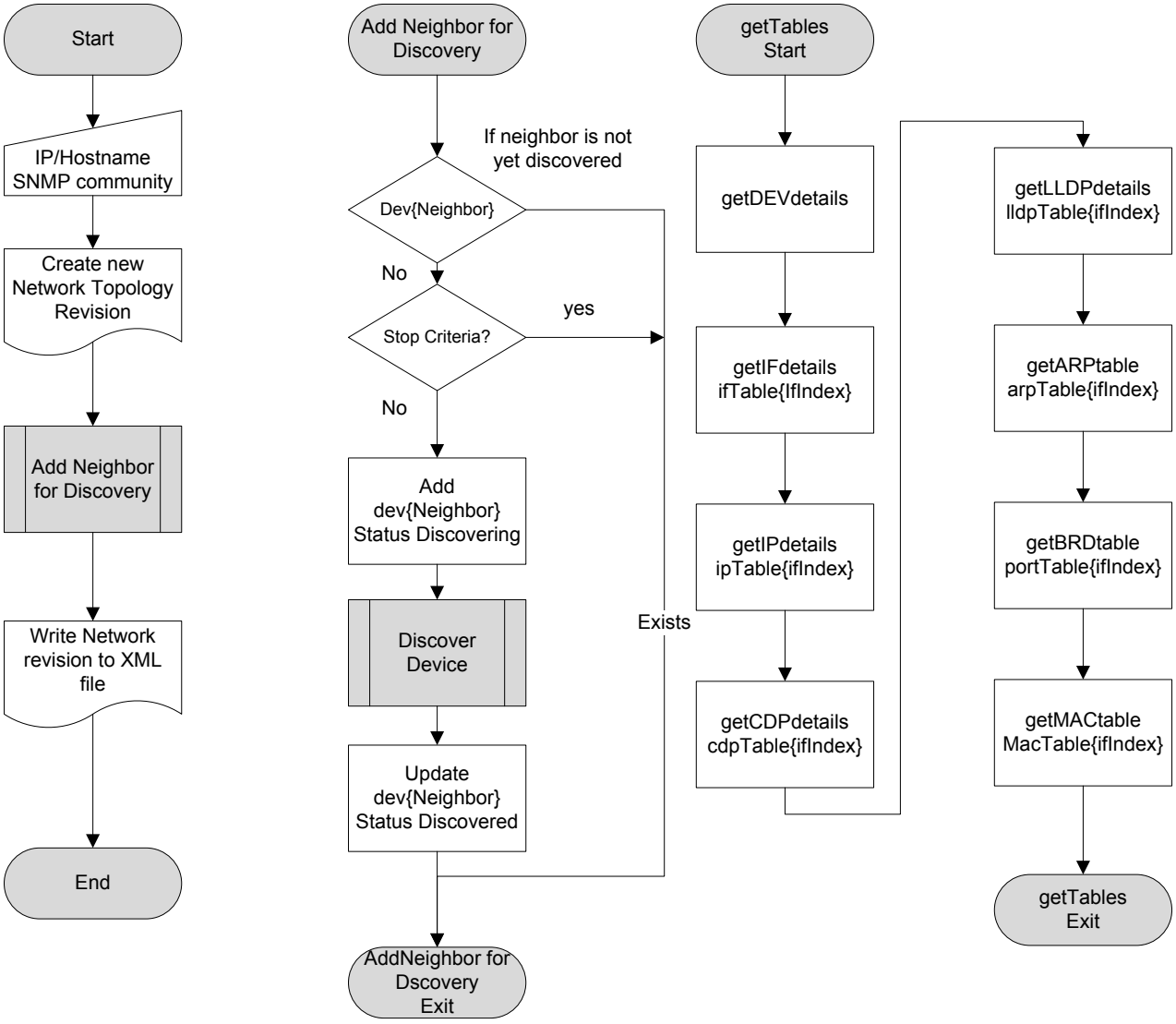


Фигура 5-2 Поведение на алгоритъма спрямо топологията на мрежата

Стъпките от алгоритъма са представени в детайли на Фигура 5-3 и

Фигура 5-4. На Фигура 5-3 е показан Main Process (основния процес) и съставните процеси „Add Neighbor for Discovery“ и „getTables”.

Фигура 5-3 Алгоритъм за разкриване на устройства (main, Add Neighbor for Discovery и getTables подпроцеси)


Start
IP/Hostname
SNMP community
Create new
Network Topology
Revision
Add Neighbor
for Discovery
Write Network
revision to XML
file
End
Add Neighbor for
Discovery
If neighbor is not
yet discovered
Dev{Neighbor}
No
Stop Criteria?
yes
No
Add
dev{Neighbor}
Status Discovering
Discover
Device
Update
dev{Neighbor}
Status Discovered
AddNeighbor for
Dscovery
Exit
Exists
getTables
Start
getDEVdetails
getIFdetails
ifTable{IfIndex}
getIPdetails
ipTable{ifIndex}
getCDPdetails
cdpTable{ifIndex}
getLLDPdetails
lldpTable{ifIndex}
getARPtable
arpTable{ifIndex}
getBRDtable
portTable{ifIndex}
getMACtable
MacTable{ifIndex}
getTables
Exit

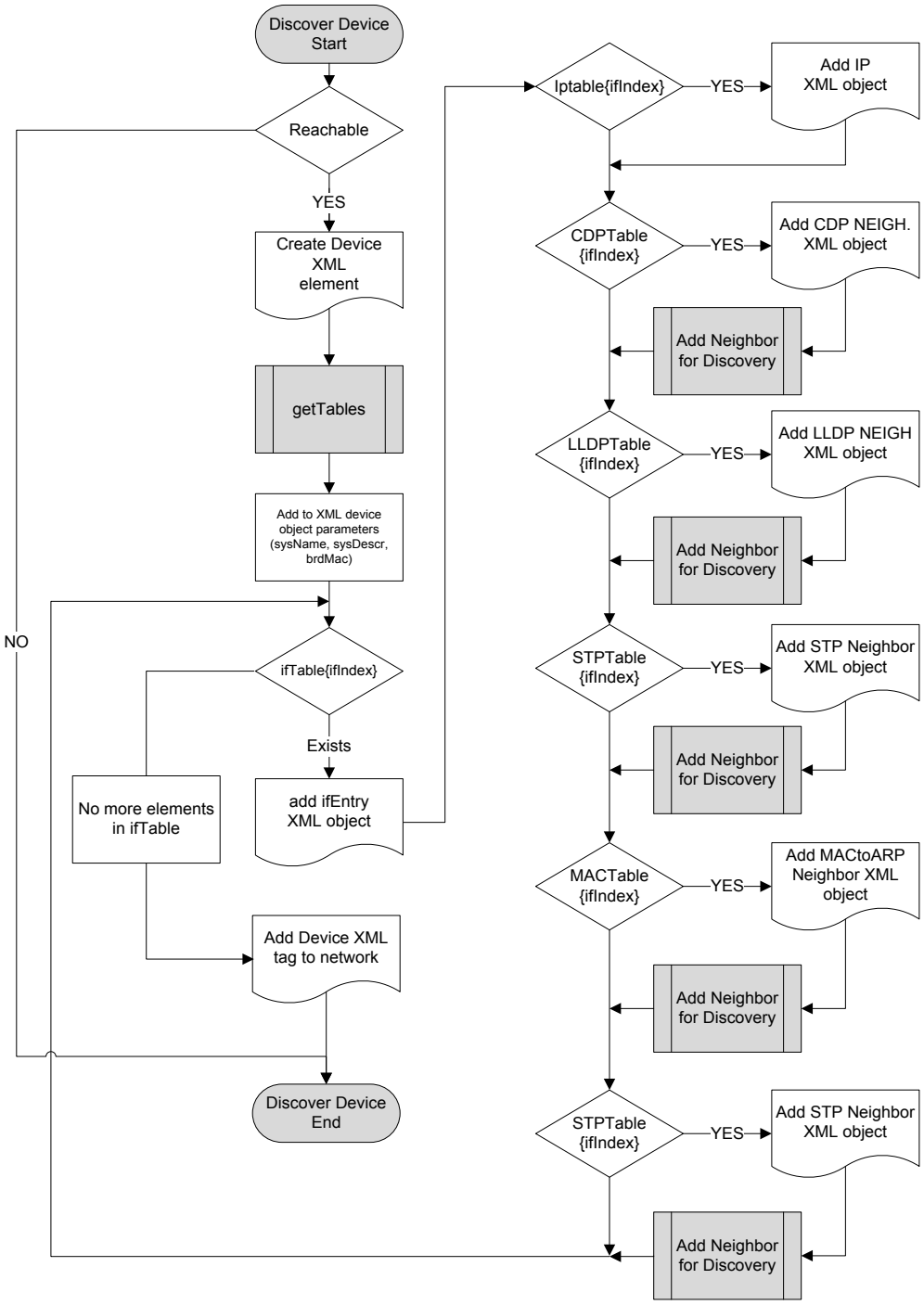
Main Process получава входящите параметри, създава нова версия на мрежовата топология и извиква процесът “Add Neighbor for Discovery”. След приключването си, процесът създава нова версия на топологията на мрежата.

Add Neighbor for Discovery проверява дали устройството не е било вече открито и дали отговаря на stop-criteria. Ако устройството не отговаря, съставният процес го добавя към таблицата от устройства за разкриване и му задава статус “discovering”. След това стартира “Discover Device” - основният процес по разкриването на устройството. След като устройството е разкрито, статусът му се променя на “discovered”.

getTables – процесът се използва за извличане на информация от разкриваното устройство. Алгоритъмът поддържа няколко метода за разкриване на съседни устройства - Cisco Discovery Protocol (CDP), Link Layer Discovery Protocol, MACtoARP, OSPF, ISIS и BGP. За всеки един от методите се генерира SNMP заявка към разкриваното устройство. Данните, получени от устройството се попълват в хеш (hash) таблици с ключ индекса на интерфейса и стойност - идентификатора на съседното устройство. За всяко едно съседно устройство се идентифицира IP адрес, име, тип и метод на разкриване.

На Фигура 5-4 е демонстриран процесът по разкриване на единично устройство „Discover Device”.

Фигура 5-4 Discovery Algorithm (Discover Device)



“Discover Device” първо проверява дали устройството е достъпно. Ако това е така, създава device-xml и изпълнява процеса getTables. Процесът getTables извлича SNMP

таблиците от устройството. След това “Discover Device” обхожда данните от ifTable. Тази таблица съдържа информация за всеки един интерфейс на разкриваното устройство. Алгоритмът построява xml структура за всеки един интерфейс. Структурите съдържат информация за IP адресите, конфигурирани на интерфейса и за потенциалните съседни устройства. За всяко едно разкрито съседно устройство се извиква процеса „Add Neighbor for Discovery process”. След като алгоритмът обходи всички интерфейси на устройството, записва данните в xml структурата на устройството и процесът приключва.

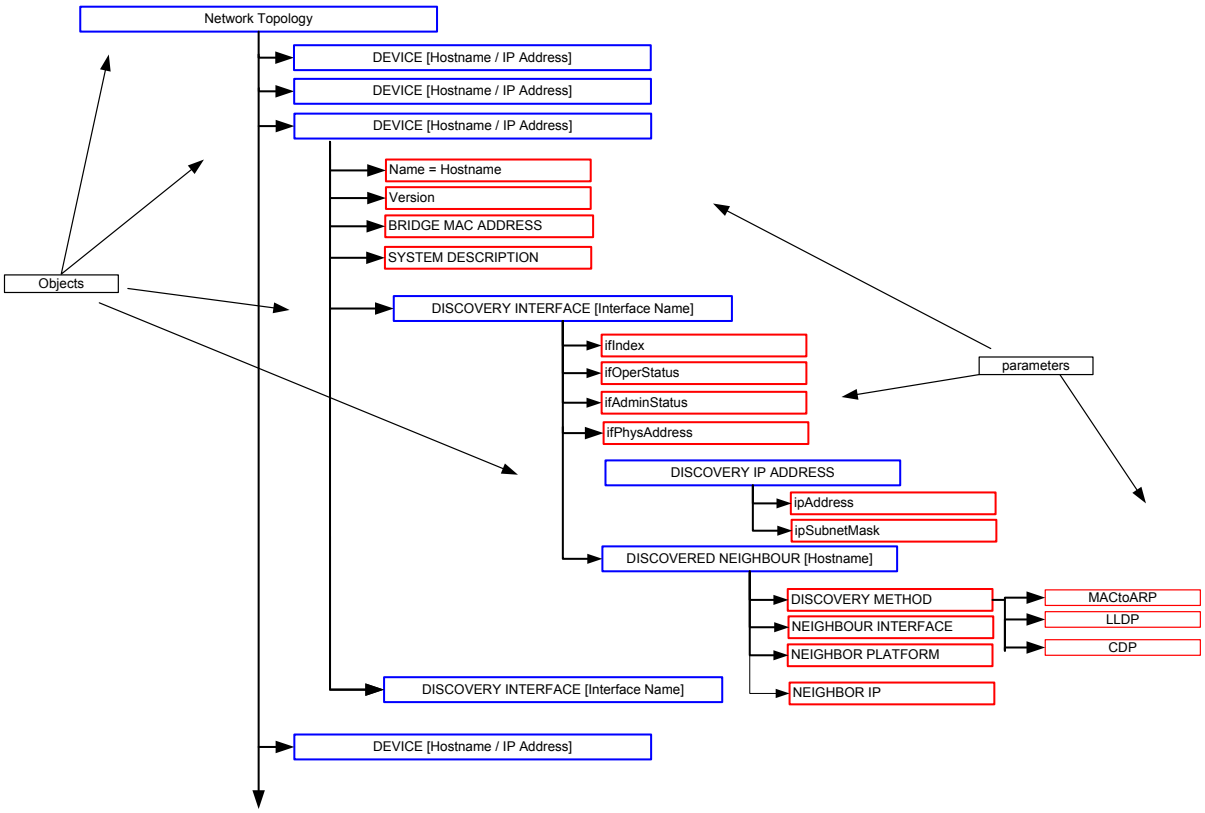
Алгоритъмът поддържа следните методите за разкриване на устройствата:

- CDP (Cisco Discovery Protocol) е протокол за разкриване на Cisco към Cisco междусъседски отношения на слой 2 от OSI модела. Въпреки че, протоколът е разработен от Cisco Systems, той се поддържа и от други производители на оборудване, включително и от HP. Методът използва данните от CDP SNMP MIB [94] [95] [96].
- LLDP (Local Link Discovery Protocol) е протокол за разкриване на междусъседски отношения в Ethernet среда, стандартизиран от IEEE. Протоколът е изключително популярен в градски Ethernet среди. Използваният от алгоритъма LLDP SNMP MIB за разкриване на устройства също е стандартизиран [97] [98] [99].
- STP (Spanning Tree Protocol) е мрежов протокол създаден с цел избягване на нежелани повторения (loop) в топологията на мрежата. Той е стандартизиран от IEEE 802.1D и се основава на алгоритъм, описан от Radia Perlman през 1985 г. Протоколът няма за цел разкриването на устройства, но тъй като създава отношения с директно свързаните съседи на ниво 2 от OSI модела, той се използва от алгоритъма за разкриване на мрежата.
- MACtoARP (Media Access Control to Address Resolution Protocol) – методът се основава на намиране на отношението между bridge, MAC и ARP SNMP таблици. В крайна сметка методът установява от кой порт е бил научен даден MAC адрес и на кое IP отговаря.

5.4 Йерархичен модел за съхранение на състоянието на мрежата

Всяко едно стартиране на алгоритъма за разкриване създава обектно - ориентиран модел (Фигура 5-5) за всяко едно от разкритите устройства. Моделът съдържа параметри на Devices (устройствата) като hostname (име), type (тип), model (хардуерен модел), Version (вид и версия на операционната система), Interface name, index, Administrative & Operation status (име на интерфейс, индекс, административен и текущ статус), Discovery IP address (открит IP адрес) и Discovered Neighbors (съседни устройства). Съседните устройства съдържат информация за: Discovery Method (методът, по който са разкрити), Neighbor Type & Name (тип и име на съседното устройство), Neighbor IP address (IP адрес на съседното устройство).

Фигура 5-5 Йерархичен модел



Йерархичният модел може да бъде съхраняван в xml структура или да бъде попълнен в йерархична база данни. Той е връзката между прототипа за трансформация на мрежата и останалите приложения от OSS/ BSS.

5.4.1 Съхранение на йерархичния, обектно-ориентиран модел в xml структура

Моделът е специфициран в XML и притежава йерархична, обектно-ориентирана структура. Всеки един обект има име, тип и се характеризира с определени параметри. Всеки един параметър има име и стойност. XML структурата на модела е демонстрирана на Фигура 5-6.

Фигура 5-6 Йерархичен, обектно ориентиран модел

```
<object>
        <name>some name</name>
        <ObjectType>some class</ObjectType>
        <parameters>
                <parameter>
                        <name>parameter name</name>
                        <value>parameter value</value>
                </parameter>
                .
                .

        </parameters>
        <object>
                .
                .
        </object>
        .
        .
</object>
```

На Фигура 5-7 е представена XSD схемата на модела. Схемата се използва за валидиране на данните от разкритата мрежа.

Фигура 5-7 XSD схема на йерархичния, обектно ориентиран модел

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="parameter">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="name"/>
        <xs:element ref="value"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="name" type="xs:string"/>
  <xs:element name="value" type="xs:string"/>
```

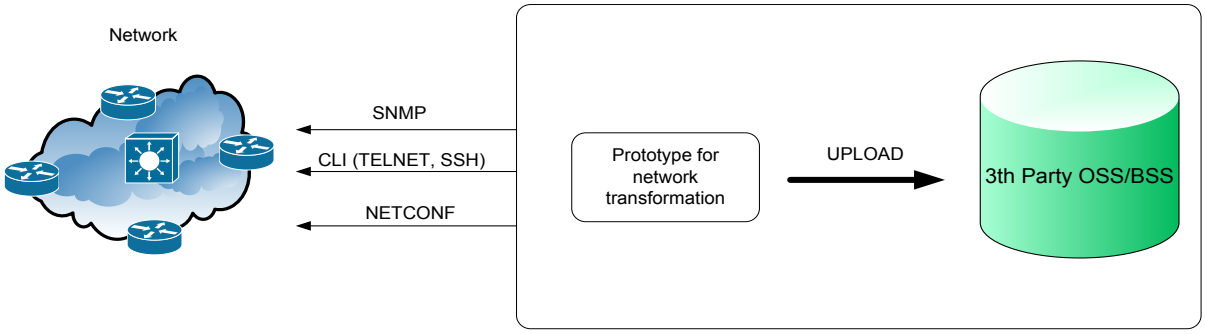
```
  <xs:element name="object">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="name"/>
        <xs:element ref="objectType"/>
        <xs:element ref="parameters"/>
        <xs:element ref="object" maxOccurs="unbounded" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="parameters">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="parameter" maxOccurs="unbounded" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="objectType">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="Discovery Interface"/>
        <xs:enumeration value="IPv4 Address"/>
       <xs:enumeration value="IPv6 Address"/>
        <xs:enumeration value="Discovered Neighbor"/>
        <xs:enumeration value="DiscoveredDevice"/>
        <xs:enumeration value="Network"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:schema>
```

5.4.2 Съхранение на йерархичния, обектно-ориентиран модел в релационна структура

Релационните бази данни са удобен вариант за съхранение на моделите, дефинирани в OSS стандарти като SID и OSS/J. Подходящи са също така и за данните от йерархичния, обектно-ориентиран модел. Голямата част от OSS/ BSS приложенията използват именно релационни бази данни за съхранение на данните за управляваните от

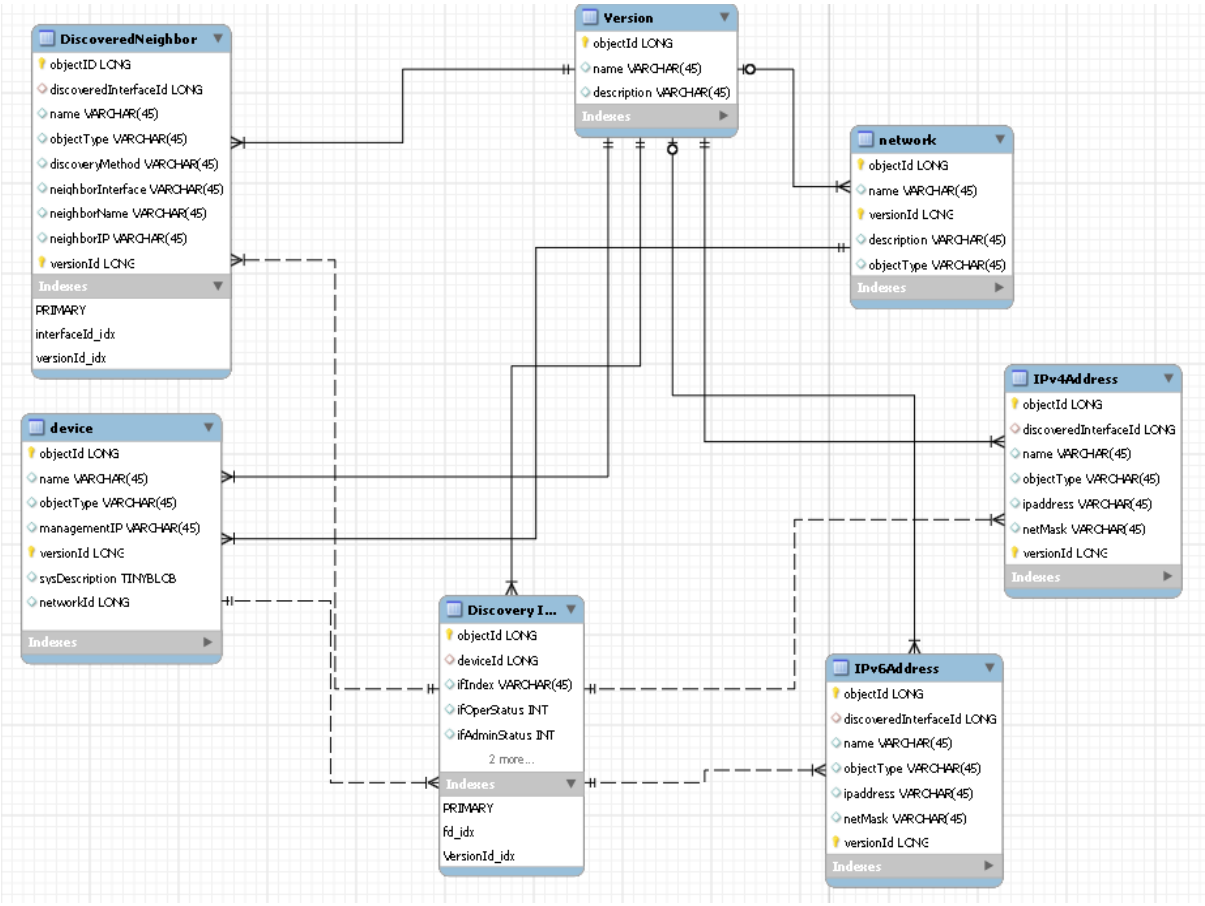
тях мрежи. Релационният модел е моста между прототипа за еволюция на мрежата от IPv4 към IPv6 и останалите OSS/ BSS системи.

Фигура 5-8 Връзка между прототипа за трансформация на мрежи и външни OSS/BSS приложения



Моделът на релационна структура, подходяща за съхранение на данните от йерархичния, обектно-ориентиран модел е показан на Фигура 5-9.

Фигура 5-9 Реализация на йерархичния, обектно ориентиран модел в релационна база данни



Моделът е дефиниран в SQL/DDL (Data Definition Language) и се състои от следните таблици:

- Version – използва се за контрол на версиите на модела. Всяка версия има уникален идентификатор - objectId, име – name и описание – description. Уникалните идентификатори на тази таблица играят ролята на външни ключове в останалите таблици.

Фигура 5-10 SQL синтаксис, за създаване на таблица Version

```
CREATE  TABLE IF NOT EXISTS `mydb`.`Version` (
  `versionId` MEDIUMTEXT NOT NULL ,
  `name` VARCHAR(45) NULL ,
  `description` VARCHAR(45) NULL ,
  PRIMARY KEY (`versionId`) );
```

- Network – съдържа мрежи. Всяка мрежа има уникален идентификатор (objectId), външен ключ - идентификатор на версията (versionId), име (name) и описание (description). Уникалността на мрежата се гарантира от първостепенен ключ по стойностите на полета objectId и versionId.

Фигура 5-11 SQL синтаксис за създаване на таблица Network

```
CREATE  TABLE IF NOT EXISTS `mydb`.`Network` (
  `objectId` MEDIUMTEXT NOT NULL ,
  `name` VARCHAR(45) NULL ,
  `versionId` MEDIUMTEXT NOT NULL ,
  `description` VARCHAR(45) NULL ,
  PRIMARY KEY (`objectId`, `versionId`) ,
  CONSTRAINT `versionId`
    FOREIGN KEY (`versionId` )
    REFERENCES `mydb`.`Version` (`versiontId` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
CREATE INDEX `versionId_idx` ON `mydb`.`Network` (`versionId` ASC) ;
```

- DiscoveredDevice – съдържа устройства. Всяко едно устройство се описва с определен брой параметри, дефинирани като колони. Уникалността на устройството се гарантира от двойката ключове objectId и versionId, където objectId е локален идентификатор на обектите от тип Device, а versionId е външен ключ генериран в таблица Verison. Йерархията между мрежата и устройствата под нея се постига чрез външния ключ networkId.

Фигура 5-12 SQL синтаксис за създаване на таблица DiscoveredDevice

```
CREATE  TABLE IF NOT EXISTS `mydb`.`DiscoveredDevice` (
  `objectId` MEDIUMTEXT NOT NULL ,
  `name` VARCHAR(45) NULL ,
  `managementIP` VARCHAR(45) NULL ,
  `versionId` MEDIUMTEXT NOT NULL ,
  `sysDescription` TINYBLOB NULL ,
  `networkId` MEDIUMTEXT NULL ,
  PRIMARY KEY (`objectId`, `versionId`) ,
  CONSTRAINT `networkId`
    FOREIGN KEY (`objectId` )
    REFERENCES `mydb`.`Network` (`objectId` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `versionId`
    FOREIGN KEY (`versionId` )
    REFERENCES `mydb`.`Version` (`versiontId` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
CREATE INDEX `objectId_idx` ON `mydb`.`DiscoveredDevice` (`objectId` ASC) ;
CREATE INDEX `networkId_idx` ON `mydb`.`DiscoveredDevice` (`objectId` ASC) ;
CREATE INDEX `versionId_idx` ON `mydb`.`DiscoveredDevice` (`versionId` ASC) ;
```

- DiscoveredInterface – съдържа обекти от тип интерфейс. Уникалността на интерфейса се гарантира от двойката ключове objectId и versionId, където objectId е локален идентификатор на обектите от тип DiscoveredInterface, а versionId е външен ключ, генериран в таблица Verison. Йерархията между устройствата и техните интерфейси се постига чрез външния ключ deviceId.

Фигура 5-13 SQL синтаксис за създаване на таблица DiscoveryInterface

```
CREATE  TABLE IF NOT EXISTS `mydb`.`DiscoveryInterface` (
  `objectId` MEDIUMTEXT NOT NULL ,
  `deviceId` MEDIUMTEXT NULL ,
  `ifIndex` VARCHAR(45) NULL ,
  `ifOperStatus` INT NULL ,
  `ifAdminStatus` INT NULL ,
  `ifPhysAddress` VARCHAR(45) NULL ,
  `versionId` MEDIUMTEXT NOT NULL ,
  PRIMARY KEY (`objectId`, `versionId`) ,
  CONSTRAINT `ParentId`
    FOREIGN KEY (`deviceId` )
    REFERENCES `mydb`.`DiscoveredDevice` (`objectId` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `VersionId`
    FOREIGN KEY (`versionId` )
    REFERENCES `mydb`.`Version` (`versiontId` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
CREATE INDEX `fd_idx` ON `mydb`.`DiscoveryInterface` (`deviceId` ASC) ;
CREATE INDEX `VersionId_idx` ON `mydb`.`DiscoveryInterface` (`versionId`
ASC);
```

- IPv4Address - съдържа обекти от тип IPv4Address. Уникалността на адреса се гарантира от двойката ключове objectId и versionId, където objectId е локален идентификатор на обектите от тип IPv4Address, а versionId е външен ключ, генериран в таблица Verison. Йерархията между интерфейсите и техните IPv4 адреси се постига чрез външния ключ discoveredInterfaceId.

Фигура 5-14 SQL синтаксис за създаване на таблица IPv4Address

```
CREATE  TABLE IF NOT EXISTS `mydb`.`IPv4Address` (
  `objectId` MEDIUMTEXT NOT NULL ,
  `discoveredInterfaceId` MEDIUMTEXT NULL ,
  `name` VARCHAR(45) NULL ,
  `ipaddress` VARCHAR(45) NULL ,
  `netMask` VARCHAR(45) NULL ,
  `versionId` MEDIUMTEXT NOT NULL ,
  PRIMARY KEY (`objectId`, `versionId`) ,
  CONSTRAINT `parentID0`
    FOREIGN KEY (`discoveredInterfaceId` )
    REFERENCES `mydb`.`DiscoveryInterface` (`objectId` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `versionId0`
    FOREIGN KEY (`versionId` )
    REFERENCES `mydb`.`Version` (`versiontId` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
CREATE INDEX `parentID_idx` ON `mydb`.`IPv4Address` (`discoveredInterfaceId`
ASC) ;
CREATE INDEX `versionId_idx` ON `mydb`.`IPv4Address` (`versionId` ASC) ;
```

- IPv6Address - съдържа обекти от тип IPv6Address. Уникалността на адреса се гарантира от двойката ключове objectId и versionId, където objectId е локален идентификатор на обектите от тип IPv6Address, а versionId е външен ключ, генериран в таблица Verison. Йерархията между интерфейсите и техните IPv6 адреси се постига чрез външния ключ discoveredInterfaceId.

Фигура 5-15 SQL синтаксис за създаване на таблица IPv6Address

```
CREATE  TABLE IF NOT EXISTS `mydb`.`IPv6Address` (
  `objectId` MEDIUMTEXT NOT NULL ,
  `discoveredInterfaceId` MEDIUMTEXT NULL ,
  `name` VARCHAR(45) NULL ,
  `ipaddress` VARCHAR(45) NULL ,
  `netMask` VARCHAR(45) NULL ,
  `versionId` MEDIUMTEXT NOT NULL ,
  PRIMARY KEY (`objectId`, `versionId`) ,
  CONSTRAINT `parentID`
    FOREIGN KEY (`discoveredInterfaceId` )
    REFERENCES `mydb`.`DiscoveryInterface` (`objectId` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `versionId`
    FOREIGN KEY (`versionId` )
    REFERENCES `mydb`.`Version` (`versiontId` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
CREATE INDEX `parentID_idx` ON `mydb`.`IPv6Address` (`discoveredInterfaceId`
ASC) ;
CREATE INDEX `versionId_idx` ON `mydb`.`IPv6Address` (`versionId` ASC) ;
```

- DiscoveredNeighbor - съдържа обекти от тип DiscoveredNeighbor. Уникалността на адреса се гарантира от двойката ключове objectId и versionId, където objectId е локален идентификатор на обектите от тип DiscoveredNeighbor, а versionId е външен ключ, генериран в таблица Verison. Йерархията между интерфейсите и устройствата, разкрити на тях се постига чрез външния ключ discoveredInterfaceId.

Фигура 5-16 SQL синтаксис за създаване на таблица DiscoveredNeighbor

```
CREATE  TABLE IF NOT EXISTS `mydb`.`DiscoveredNeighbor` (
  `objectID` MEDIUMTEXT NOT NULL ,
  `discoveredInterfaceId` MEDIUMTEXT NULL ,
  `name` VARCHAR(45) NULL ,
  `objectType` VARCHAR(45) NULL ,
  `discoveryMethod` VARCHAR(45) NULL ,
  `neighborInterface` VARCHAR(45) NULL ,
  `neighborName` VARCHAR(45) NULL ,
  `neighborIP` VARCHAR(45) NULL ,
  `versionId` MEDIUMTEXT NOT NULL ,
  PRIMARY KEY (`objectID`, `versionId`) ,
  CONSTRAINT `interfaceId`
    FOREIGN KEY (`discoveredInterfaceId` )
    REFERENCES `mydb`.`DiscoveryInterface` (`objectId` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `versionId`
    FOREIGN KEY (`versionId` )
    REFERENCES `mydb`.`Version` (`versiontId` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
CREATE INDEX `interfaceId_idx` ON `mydb`.`DiscoveredNeighbor`
(`discoveredInterfaceId` ASC) ;
CREATE INDEX `versionId_idx` ON `mydb`.`DiscoveredNeighbor` (`versionId` ASC)
;
```

5.5 Графовиден модел от данни за съхранение на състоянието на мрежата

Графовидният модел е базиран на Graphml – xml файлов формат, създаден за моделиране на графове. Графовидният модел се получава след трансформация на данните от йерархичния модел. Процесът на трансформация се извършва чрез прилагане на XSL (Extensible Stylesheet Language) трансформация върху xml структурата на йерархичния модел. XSL е език за трансформация на документи, стандартизиран от W3C [100].

Всеки един graphml файл съдържа един или повече от един граф, а всеки граф съдържа възли (nodes) и връзки (edges). Графът може да бъде насочен, ненасочен или смесен. В графа се дефинират свойства (data-keys), които могат да притежават възлите и връзките. Съответно всеки един възел има идентификатор, етикет и се описва с определени, предефинирани на ниво граф свойства. Връзките се описват с идентификатори, етикет, начален и краен възел и определен брой свойства. Опционално връзката може да съдържа входящ и изходящ порт. Graphml поддържа също така и hyperedges – сложна връзка между повече от два възела. На Фигура 5-17 е демонстриран graphml пример с насочен граф G (edgedefault=”directed”), съдържащ четири възела – n0, n1, n2, n3, една връзка – n0-n3 между портове “North“ и „NorthEast” и един hyperedge между n0, n1 и n2 с портове North, East и Southeast.

Фигура 5-17 Graphml файл формат

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <graph id="G" edgedefault="directed">
    <node id="n0">
      <port name="North"/>
      <port name="South"/>
      <port name="East"/>
      <port name="West"/>
    </node>
    <node id="n1">
      <port name="North"/>
      <port name="South"/>
      <port name="East"/>
      <port name="West"/>
    </node>
    <node id="n2">
      <port name="NorthWest"/>
      <port name="SouthEast"/>
    </node>
    <node id="n3">
```

```
      <port name="NorthEast"/>
      <port name="SouthWest"/>
    </node>
    <edge source="n0" target="n3" sourceport="North" targetport="NorthEast"/>
    <hyperedge>
       <endpoint node="n0" port="North"/>
       <endpoint node="n1" port="East"/>
       <endpoint node="n2" port="SouthEast"/>
     </hyperedge>
  </graph>
</graphml>
```

Една от важните причини за избор на graphml, като формат за съхранение на състоянията на мрежата, е възможността му за контролирани разширения на база добавяне на нови атрибути към съществуващите xml елементи.

Пример за подобно разширение е добавянето на xlink атрибут към съществуващите възли. Това би позволило съхранение на свойствата на възела на произволно място в Интернет. Например в контекста на настоящия проект това би позволило на всеки възел да има xlink референция към йерархичния обектно ориентиран модел.

Фигура 5-18 Добавяне на атрибут, съдържащ референция към йерархичния, обектно ориентиран модел

```
<node id="n0" xlink:href="http://itransformer.com/device-xml-no.xml"/>
```

Разширението се дефинира като допълнение към съществуваща xsd схема.

Фигура 5-19 xlink разширение на xsd схемата на graphml файлов формат

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
   targetNamespace="http://graphml.graphdrawing.org/xmlns"
   xmlns="http://graphml.graphdrawing.org/xmlns"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   elementFormDefault="qualified"
   attributeFormDefault="unqualified"
>

<xs:import namespace="http://www.w3.org/1999/xlink"
           schemaLocation="xlink.xsd"/>
```

```
<xs:redefine
    schemaLocation="http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <xs:attributeGroup name="node.extra.attrib">
    <xs:attributeGroup ref="node.extra.attrib"/>
    <xs:attribute ref="xlink:href" use="optional"/>
  </xs:attributeGroup>
</xs:redefine>

</xs:schema>
```

5.6 Визуализация на топологията на мрежата

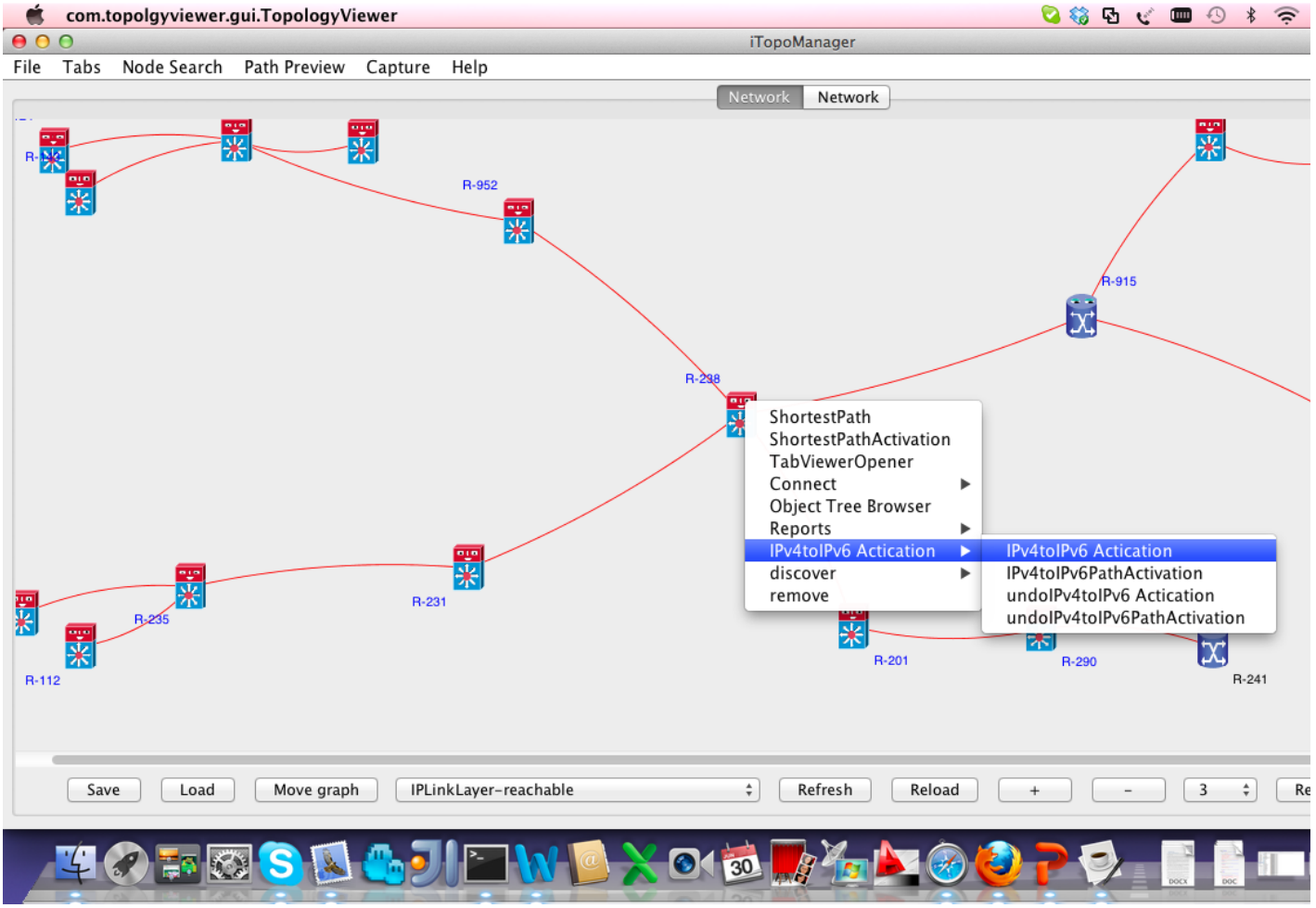
Визуализацията на топологията на мрежата е сред важните изисквания към прототипа за трансформация на мрежата. Според изискванията, топологията трябва да може да се филтрира на база на свойствата на възлите и връзките в нея.

За реализацията на тези изисквания е използван JUNG (Java Universal Network/Graph) [101]. JUNG е библиотека с отворен код, предоставяща абстрактни интерфейси за работа с графовидни, мрежови структури. Библиотеката позволява манипулации на графа, филтрация, анализи и визуализация.

На базата на JUNG е създаден графичния интерфейс на прототипа за трансформация на мрежата, демонстриран на Фигура 5-20.

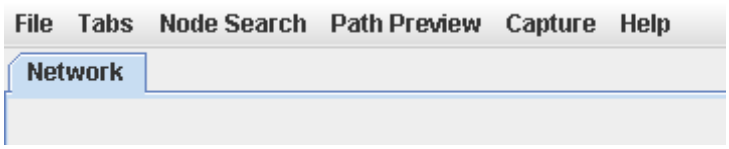
Всеки един възел е визуализиран с конкретна иконка. Изборът на иконите е според свойствата на възлите. За целта са използвани свойства като deviceModel и deviceType. Връзките също са визуализирани с конкретни линии спрямо конкретни комбинации от свойства, които те притежават. Например връзките от градския метро сегмент са показани със сини линии, а връзките от опорната MPLS мрежа с червени.

Фигура 5-20 Графичен Интерфейс



Прототипът поддържа следната функционалност в основното си меню (горе вляво).

Фигура 5-21 Main Menu



File (Файлово Меню)

- Open – отваря състояние на мрежата от файловата структура.
- Open Remote – зарежда състояние на мрежата от URL на отдалечен сървър.
- Config – зарежда конфигурационен файл.
- Diff – сравнява две състояния на мрежовия граф и показва разликите.
- Remote Config – позволява зареждането на конфигурационния файл от отдалечен сървър.
- Initial Node – използва се при визуализации на големи мрежи. Ако бъде зададен, топологията се изчертава на определен брой възли от първоначалния възел.

Tabs (Меню за отваряне на нови полета)

- New Tab – отваря ново поле с топологията на мрежата. Това позволява прилагането на различни филтри върху една и съща топология в различните прозорци.
- Close – затваря полето.
- Close All – затваря всички отворени полета.

Node Search (Търсене на възел)

- Search by Name Current Graph - търсене на възел в текущия (визуализирания на екрана) граф.
- Search by Name Entire Graph – търсене на възел в целия граф.
- Search by key – търсене по свойство на възел.
- Search by IP – търсене по IP адрес.

Path Preview (Търсене на път)

- Shortest Path - търсене на най-краткия път между два възела в мрежата.
- Weighted Shortest Path – търсене на най-краткия път на най-ниска цена.

Capture (Запис на топологията като изображение)

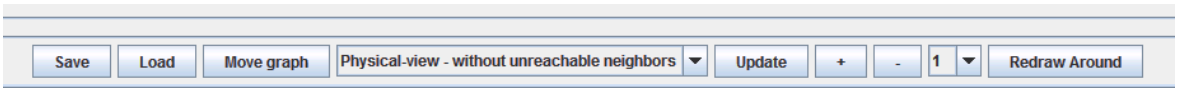
- Capture to PNG – запис на топологията в PNG файл формат.
- Capture to EPS – запис на топологията като EPS файл формат.

Help

- Информация за текущата версия.
- User Guide – препратка към потребителската документация.
- About US – информация за авторския колектив.

Графичният интерфейс позволява работа със заредения граф чрез централно разположеното меню в долната част на прозореца (Фигура 5-22).

Фигура 5-22 Бутони в панела на TopologyViewer



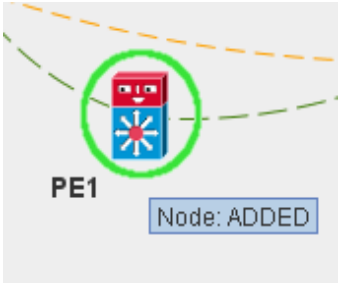
- Save – запазва подредбата на графа във файл.
- Load – зарежда подредения граф от файл.
- Move graph – позволява придвижване на цялата топология във видимото поле на екрана.
- Filters selection – позволява избор на различни филтри на информацията на екрана. Филтрите са дефинирани в конфигурационния файл.
- Update - обновява текущата топология.
- Plus/Minus buttons – увеличава/ намалява изображението.
- Redraw around – позволява изчертаване на топологията на определен брой възли от избрания възел.

5.7 Визуализация на разликите между текущото и предходното състояние

Разликите между две състояния могат да се изразяват в следното:

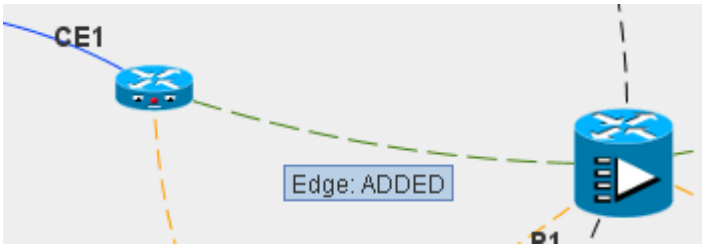
- Поява на нови възли - появата на нов възел се отбелязва със съответната икона за възела и добавяне на кръг в зелен цвят около иконата (Фигура 5-23). При преминаване с курсора на мишката върху иконата се появява информационно поле, съдържащо надпис “Node: Added”. В текущото състояние на мрежата има нов възел.

Фигура 5-23 Поява на нов възел



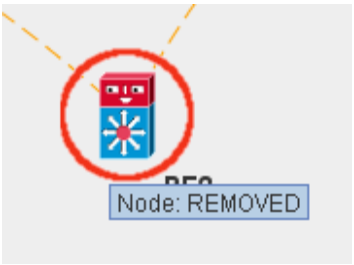
- Поява на нови връзки - новите връзки се обозначават с пунктирана линия в зелено (Фигура 5-24). При преминаване с курсора на мишката върху иконата се появява информационно поле съдържащо надпис “Edge: Added”. В текущото състояние на мрежата има нова връзка.

Фигура 5-24 Поява на нова връзка



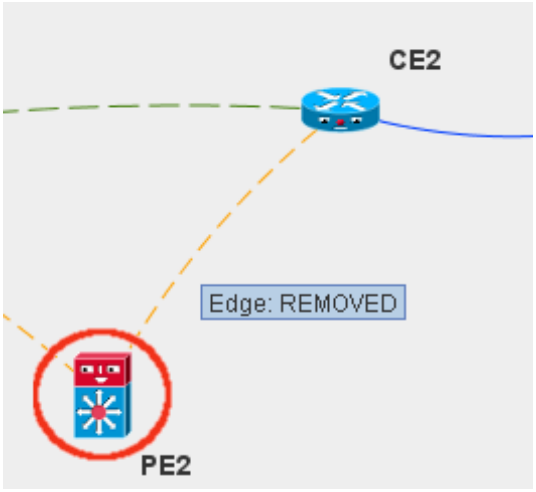
- Липса на възли - липсата на възел в текущото състояние се отбелязва с червена окръжност около иконата на възела (Фигура 5-25) . При преминаване с курсора на мишката върху иконата се появява информационно поле съдържащо надпис “Node: Removed”. В текущото състояние липсва възел.

Фигура 5-25 Липса на възел



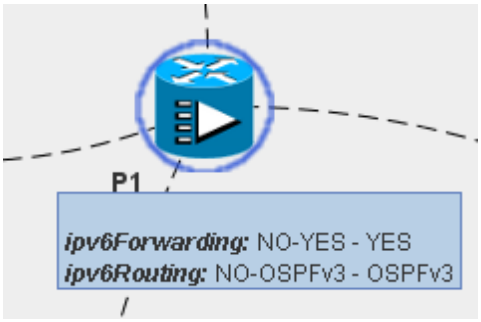
- Липса на връзки - липсата на връзки в текущото състояние се отбелязва с пунктирана линия с оранжев цвят (Фигура 5-26). При преминаване с курсора на мишката върху иконата се появява информационно поле, съдържащо надпис “Edge: REMOVED”. В текущото състояние на мрежата липсва дадената връзка.

Фигура 5-26 Липса на връзка



- Промяна на съществуващите свойства на възел - промените в свойствата на даден възел се визуализират чрез синя окръжност около иконата на възела. Конкретните промени стават видими при преминаване с курсора на мишката върху иконата. Тогава се появява информационно поле, съдържащо информация за конкретните свойства, които са били променени.

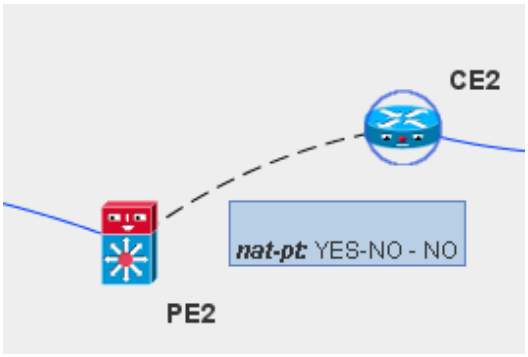
Фигура 5-27 Промяна свойствата на възел



- Промяна на съществуващите свойства на връзката - промените в свойствата на дадена връзка се визуализират чрез черна пунктирана линия. Конкретните промени стават видими при преминаване с курсора на мишката през

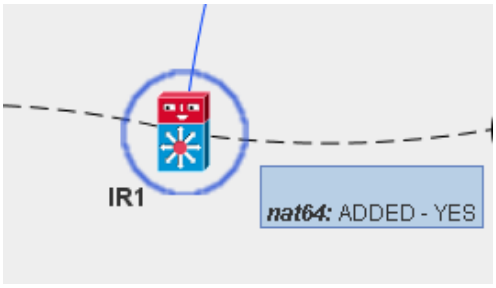
конкретната връзка. Тогава се появява информационно поле, съдържащо информация за конкретните свойства, които са претърпели промяна.

Фигура 5-28 Промяна свойствата на връзка



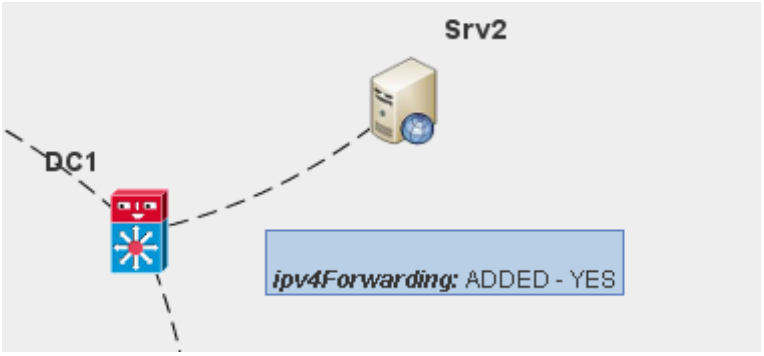
- Добавяне на свойства на възел - възлите не само могат да претърпят промяна на дадени свойства, но и могат да придобият нови такива в процеса на еволюция на дадена мрежа. В тези случаи възелът се визуализира със синя окръжност около иконата. Конкретното свойство, което е добавено става видимо при преминаване с мишката над дадената иконка.

Фигура 5-29 Добавяне на свойства на възел



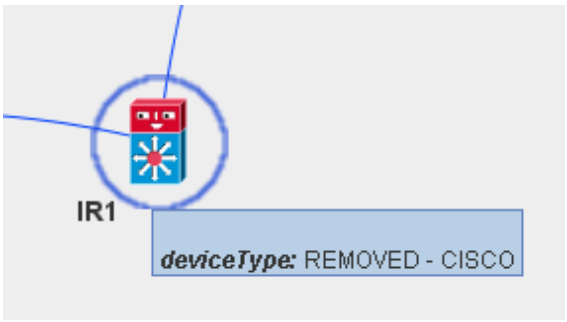
- Добавяне на свойства на връзка - връзките също могат да придобият нови свойства в процеса на еволюция на дадена мрежа. В тези случаи връзката се визуализира с черна пунктирана линия. Конкретното свойство, което е добавено става видимо при преминаване с мишката над дадената иконка.

Фигура 5-30 Придобиване на нови свойства в текущото състояние



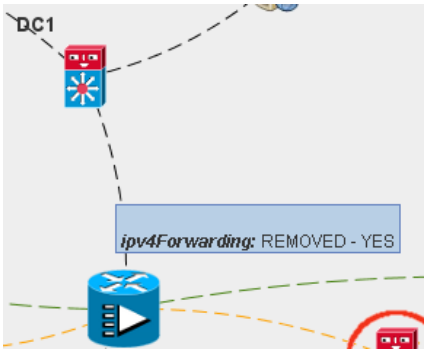
- Загуба на свойства на възел - възлите могат не само да придобият нови свойства, но и да загубят такива в процеса на еволюция. В тези случаи възелът се визуализира чрез синя окръжност около иконката, а конкретните свойства стават видими за потребителя при преминаване с мишката над иконката на конкретния възел (Фигура 5-31).

Фигура 5-31 Загуба на свойства на възел в текущото състояние



- Загуба на свойства на връзка - връзките, също както и възлите, могат не само да придобият нови свойства, но и да загубят такива в процеса на еволюция. В тези случаи връзката се визуализира чрез черна пунктирана линия, а конкретните свойства стават видими за потребителя при преминаване с мишката над иконката на конкретната връзка (Фигура 5-32).

Фигура 5-32 Загуба на свойства на връзка

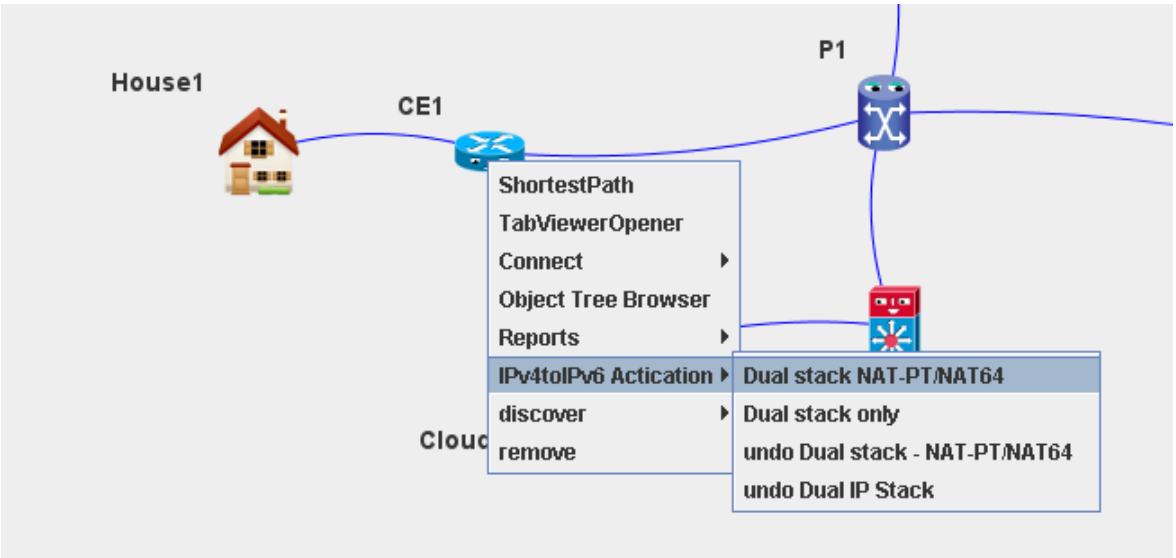


5.8 Изпълнение на стъпките от стратегията

Прототипът на системата за трансформация на мрежи от IPv4 към IPv6 съдържа модел на стъпките от стратегията, избрана за еволюционен път и може да ги изпълнява на устройствата в мрежата. При избор на стъпка се изпълнява следната последователност от действия:

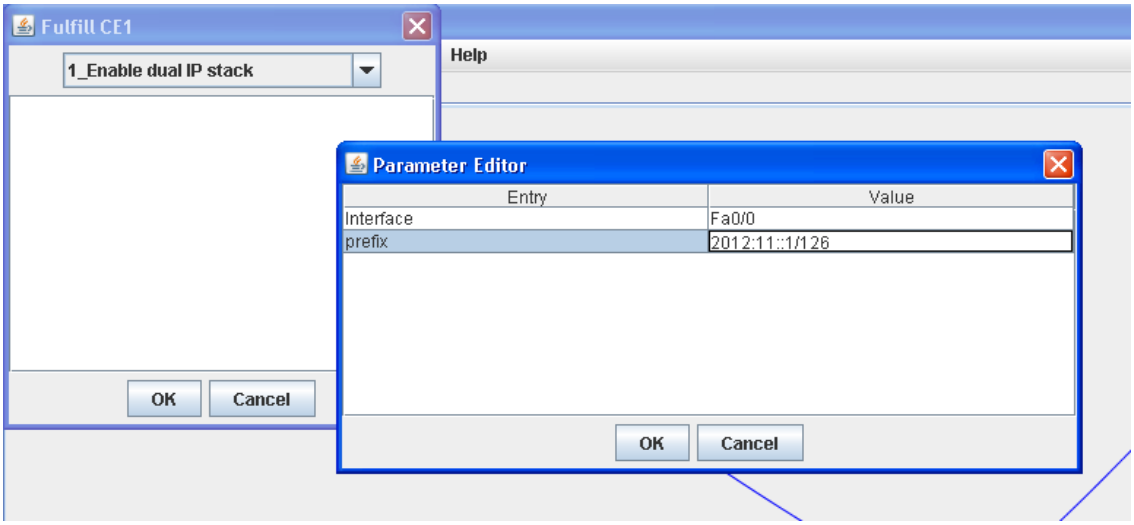
5.8.1 Извикване на стъпка от стратегия

Фигура 5-33 Извикване на стъпка



5.8.2 Подаване на входящи параметри

Фигура 5-34 Подаване на входящи параметри



Параметрите може да бъдат подадени чрез въвеждане във входяща форма, да бъдат извлечени автоматично от текущия граф или от йерархичния модел. Описанието на параметрите, необходими на дадена стъпка се извършва чрез описание на XML структура, наречена paramFactory (Фигура 5-35). Нейната цел е да създава групи от параметри и да извлича техните стойности от различни източници на данни.

Фигура 5-35 parameFactory

```
      <param-factory name="ipv6Interface">
       <param-factory-element type="manual">
            <param name="prefix"/>
            <param name="Interface"/>
        </param-factory-element>
        <param-factory-element type="graphml">
            <param name="ManagementIPAddress"/>
            <param name="hostname"/>
        </param-factory-element>
        <param-factory-element type="resource">
            <param name="username"/>
            <param name="password"/>
            <param name="enable-password"/>
        </param-factory-element>
    </param-factory>
```

В примера на Фигура 5-35 са дефинирани три типа параметри – prefix и interface се въвеждат през автоматично генерирана форма, ManagementIPAddress и hostname се извличат от graphml модела, а username, password и enable-password от xml, дефиниращ параметрите за комуникация с ресурсите. ResourceXml описва параметрите на протоколите, чрез които прототипът комуникира с мрежата (Фигура 5-36).

Фигура 5-36 Дефиниция на параметри за комуникация с физическите ресурси на мрежата

```
<resource name="cisco">
        <param name="deviceType">CISCO</param>
        <connection-params connection-type="telnet">
<param name="username">username</param>
              <param name="password">password</param>
              <param name="enable-password">password</param>
              <param name="timeout">3000</param>
              <param name="retries">3</param>
        </connection-params>
        <connection-params connection-type="snmp">
              <param name="community-ro">public</param>
              <param name="community-rw">private</param>
              <param name="timeout">1000</param>
              <param name="retries">3</param>
```

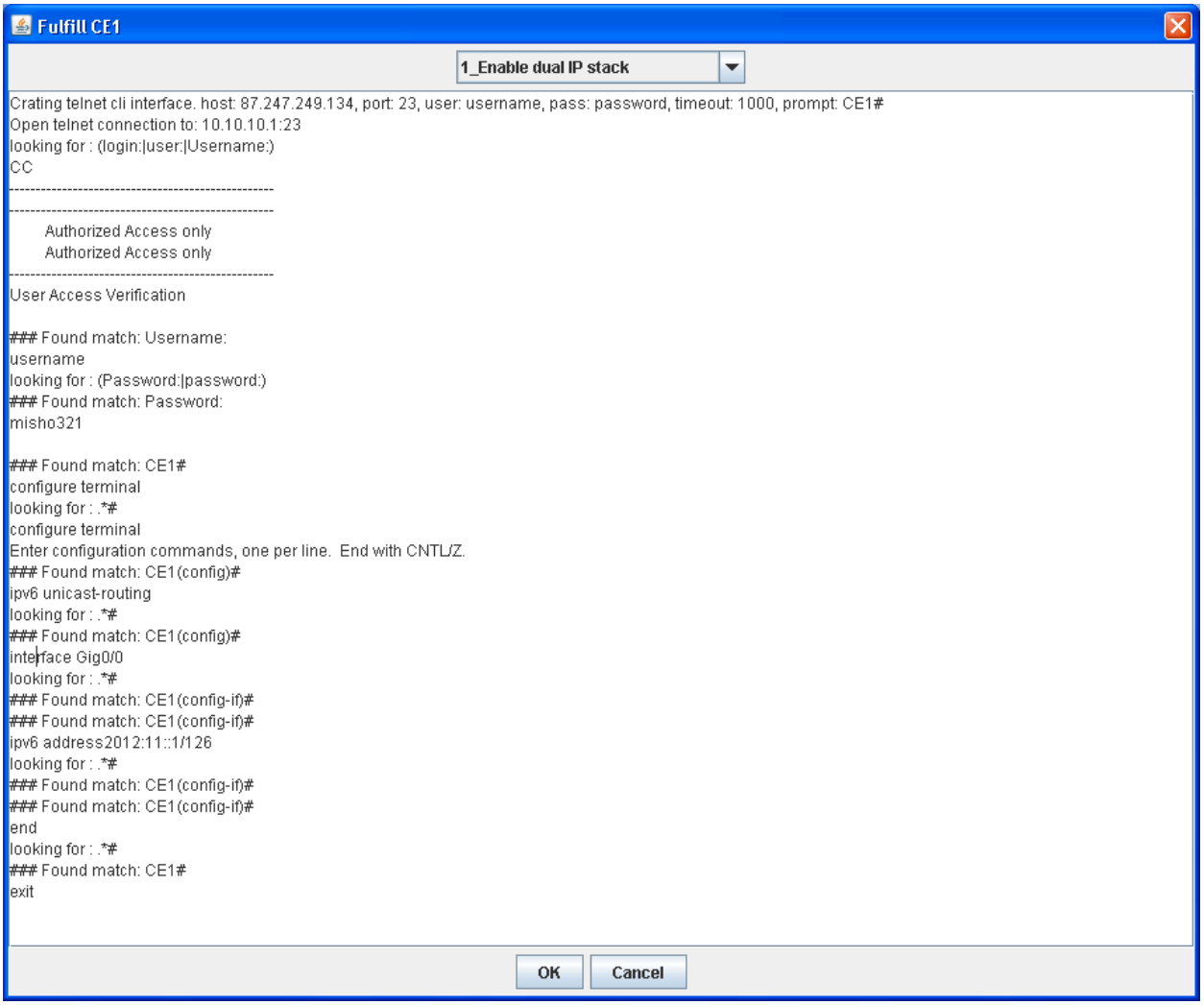
```
        </connection-params>
    </resource>
```

5.8.3 Изпълнение на проверките от техническите ограничения

Ако техническите ограничения към стъпката са спазени, се преминава към следващата стъпка. Ако не са изпълнени, се прекратява изпълнението на стъпката.

5.8.4 Изпълнение на действието

Фигура 5-37 Изпълнение на действието

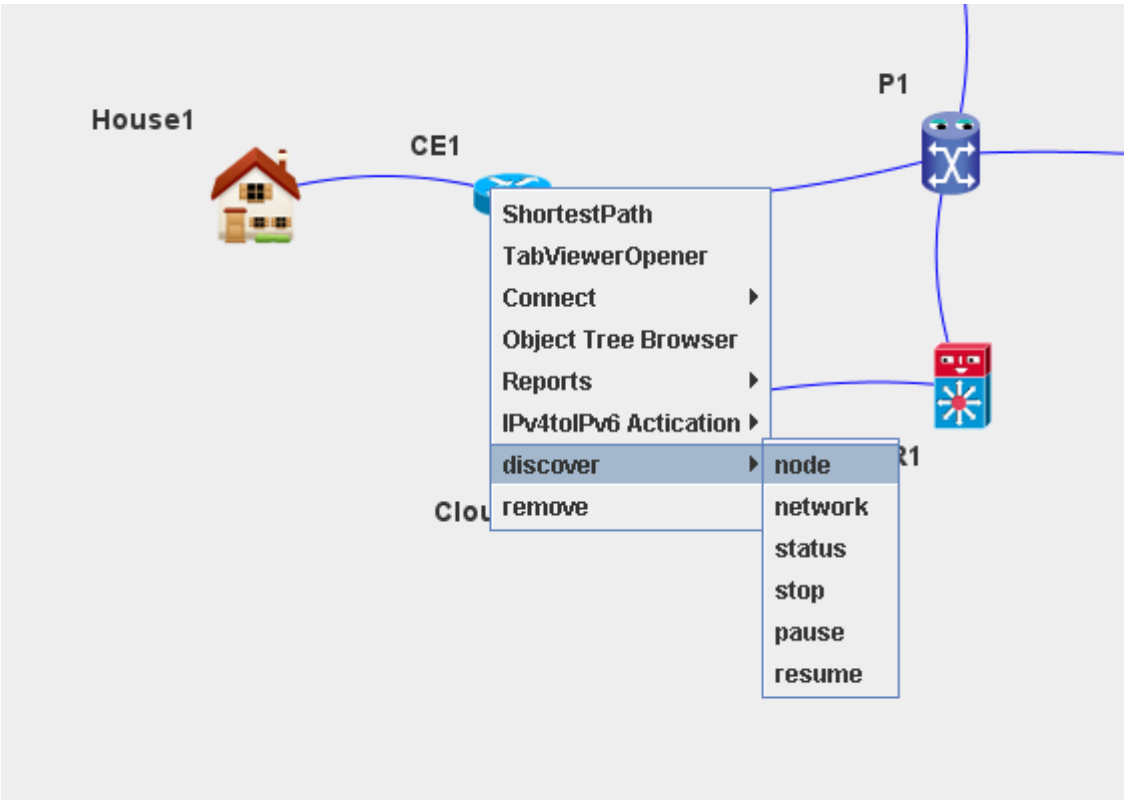


След като параметрите бъдат подадени, следва изпълнението на шаблона от команди, характеризиращи действието в стъпката (Фигура 5-37).

5.8.5 Разкриване на текущото състояние на мрежата

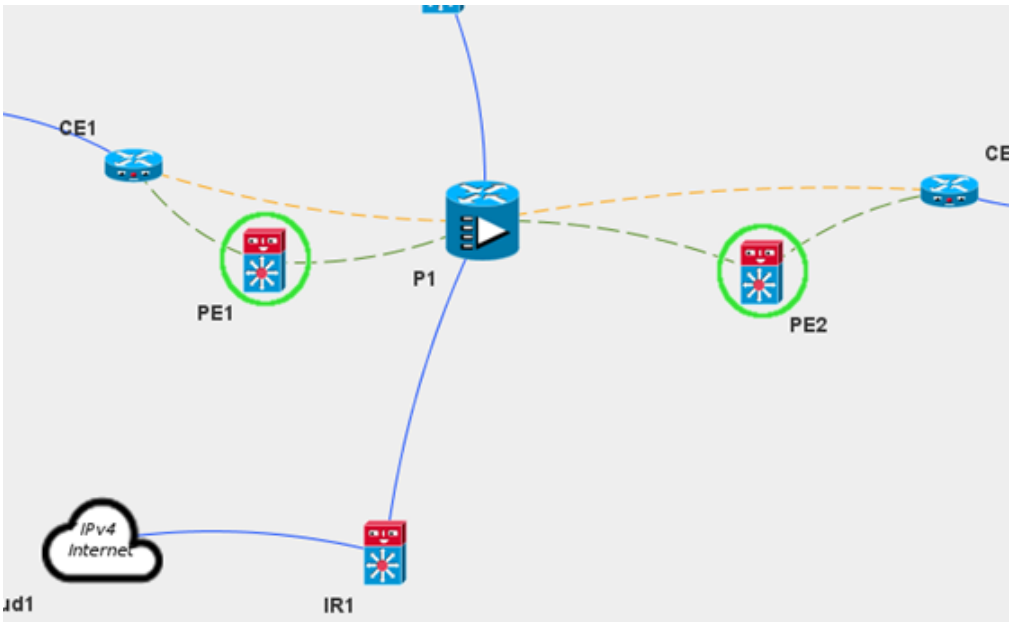
Процесът на повторно разкриване на мрежата се стартира само за възлите, чиято конфигурация е била променена при изпълнение на действието. Стартирането става чрез селектиране на тези възли и извикване на меню с десен бутон на мишката. В появилото се меню се избира опцията discover->node (т.е. разкрий устройство) (Фигура 5-38).

Фигура 5-38 Стартиране на процеса по повторното разкриване на мрежата



5.8.6 Представяне на разликите между първоначалния модел на мрежата и модела на текущото състояние

Фигура 5-39 Представяне на разликите между предходното и текущото състояние



На Фигура 5-39 е демонстриран пример, при който са добавени две нови устройства (PE1 и PE2) и трафика между CE1/2 и P1 е пренасочен през новите устройства.

5.8.7 Проверка на ефекта

Проверката на ефекта се изпълнява върху текущото състояние на мрежата, генерирано след трансформацията на мрежата. Проверката се състои в заявка към графовидния модел, проверяваща дали са налични или не определени възли, връзки или свойства на възлите/връзките.

5.8.8 Изпълнение на обратната стъпка

Ако текущото състояние не съдържа необходимите промени и проверката на ефекта е неуспешна, потребителят има възможност да стартира предварително дефинирана “rollback” стъпка и да върне мрежата в предходното ѝ състояние.

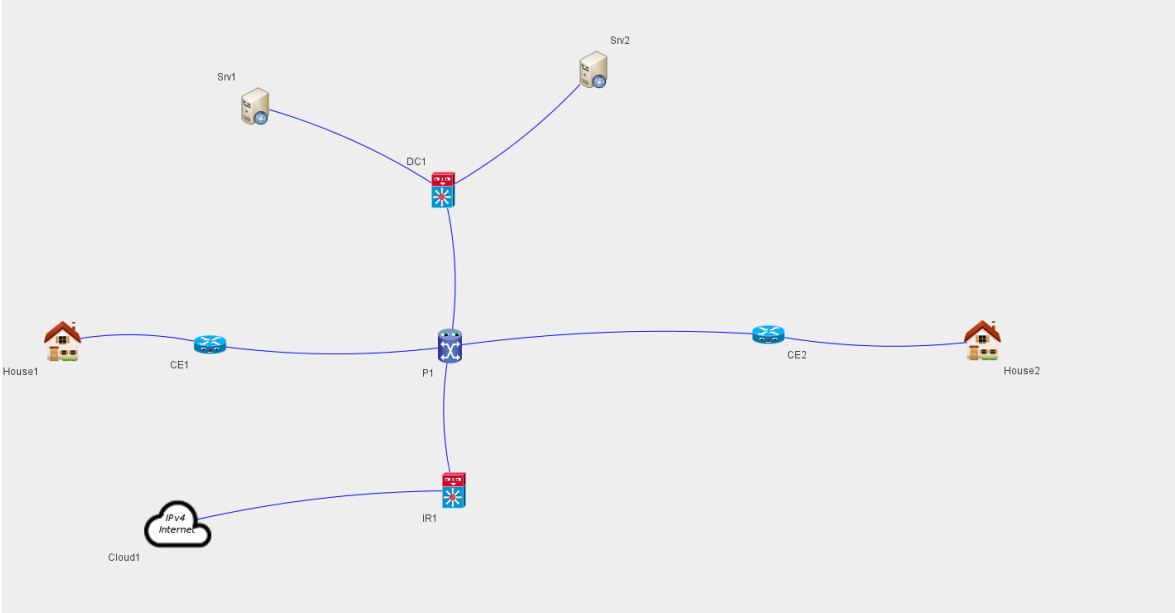
5.9 Еволюция на мрежата на оператор X от състояние “IPv4 only” до състояние “IPv6 only”

Еволюционният път ще бъде извървян по стъпките от стратегията „Преход към IPv6 чрез превод на адреси и двоен IP стек“.

5.9.1 Първоначално състояние (IPv4 Only)

“IPv4 Only” е първоначалното (текущо) състояние на мрежата.

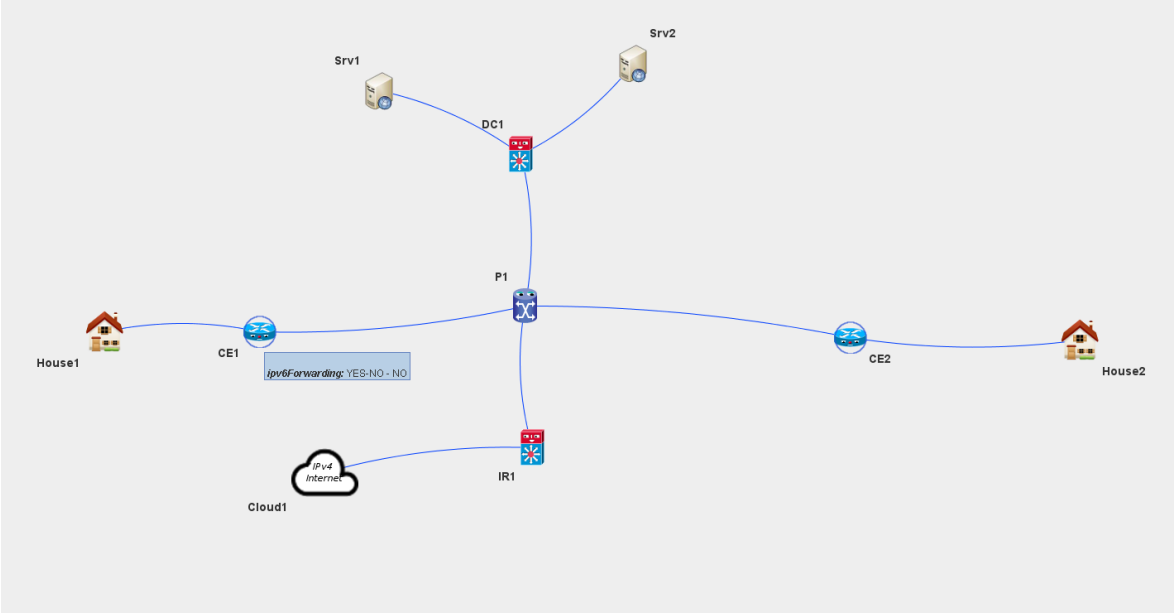
Фигура 5-40 Първоначално състояние на мрежата (IPv4 Only)



5.9.2 CE IPv6 Capable

Състоянието се достига след изпълнение на стъпка „Enable dual IP stack on CE“ на устройства CE_1 и CE_2. Разликите между “IPv4 Only” и текущото състояние са във възли CE1 и CE2. При изпълнението на стъпката те са били конфигурирани да поддържат IPv6 unicast-routing и предаване на IPv6 пакети на ниво интерфейс. Според допусканията в първоначалното си състояние CE1 и CE2 са били със свойство ipv4Forwarding=“YES” и с тази промяна те са придобили и свойство ipv6Forwarding=“YES”. Новото свойство е било разкрито чрез извличане на SNMP стойността на OID (Object Identifier) “ipv6Forwarding”.

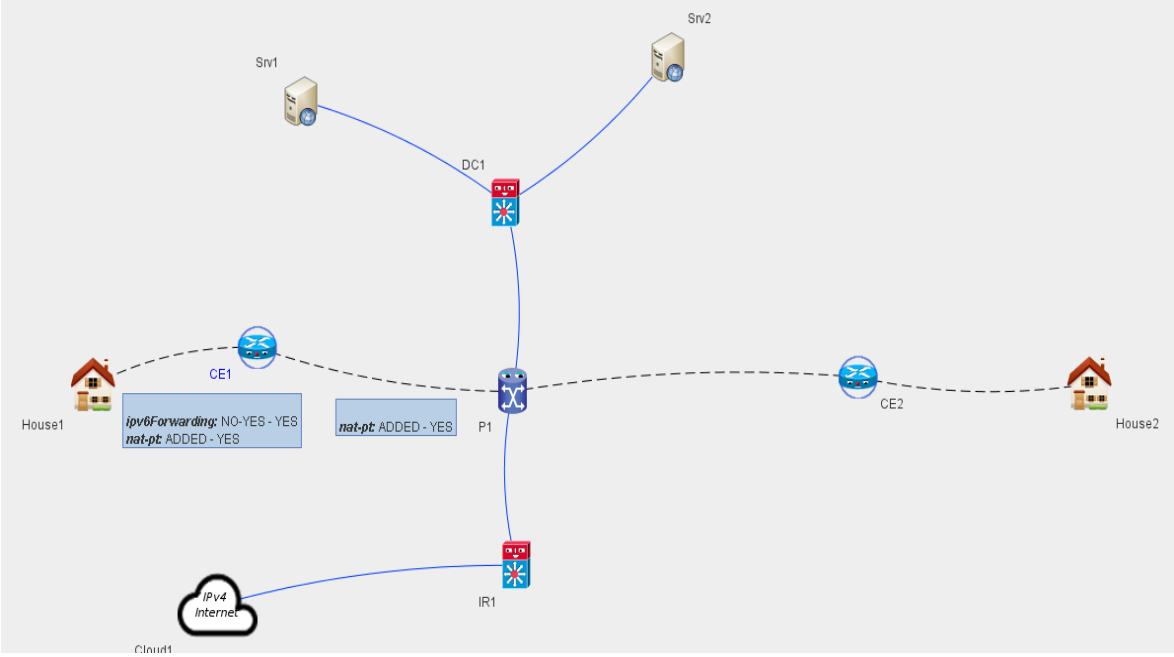
Фигура 5-41 Сравнението между първоначалното (IPv4 Only) и текущото (CE IPv6 Capable)



5.9.3 CE able to translate IPv6 to IPv4

Състоянието се достига след изпълнение на стъпка „Enable NAT-PT“ на CE_1 и CE_2, като двойният IP стек трябва да бъде конфигуриран на интерфейса към интелигентния дом (House1/House2), а NAT-PT на интерфейсите, водещи към интелигентния дом и към останалата част на мрежата.

Фигура 5-42 Сравнение между състояния “CE IPv6 Capable” и “CE able to translate IPv6 to IPv4”

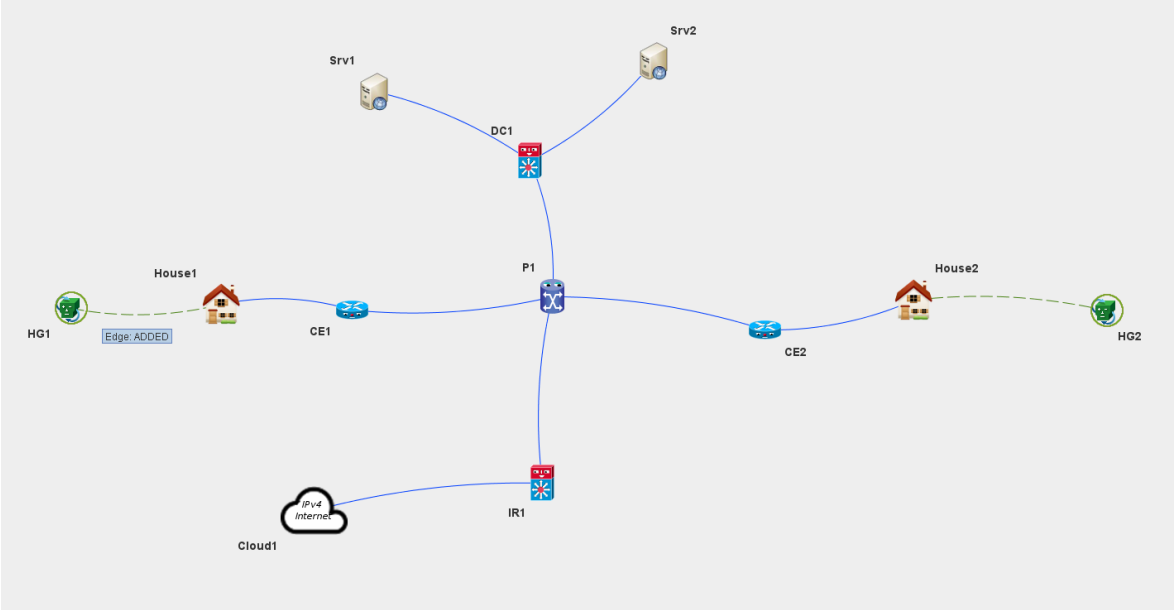


Разликите в сравнение с предишното състояние са във възли CE1 и CE2 и във връзките между тях и интелигентния дом (House 1, House2) и P1. CE1 и CE2 са придобили свойство nat-pt=„YES”, връзките между House1, House2 и CE1, CE2 са придобили свойства ipv6Forwarding=“YES” и nat-pt=“YES“, връзките между CE1,2 и P1 - свойство YES.

5.9.4 Building Automation in Production

Състоянието се достига след изпълнението на стъпка “Add HG”. Добавянето на домашен шлюз ефективно стартира процеса по таксуване за услугата „Интелигентен дом“. Стъпката по добавянето на HG в реалния свят се състои в монтаж на устройството в сградата на клиента и свързването му към съществуващия домашен рутер.

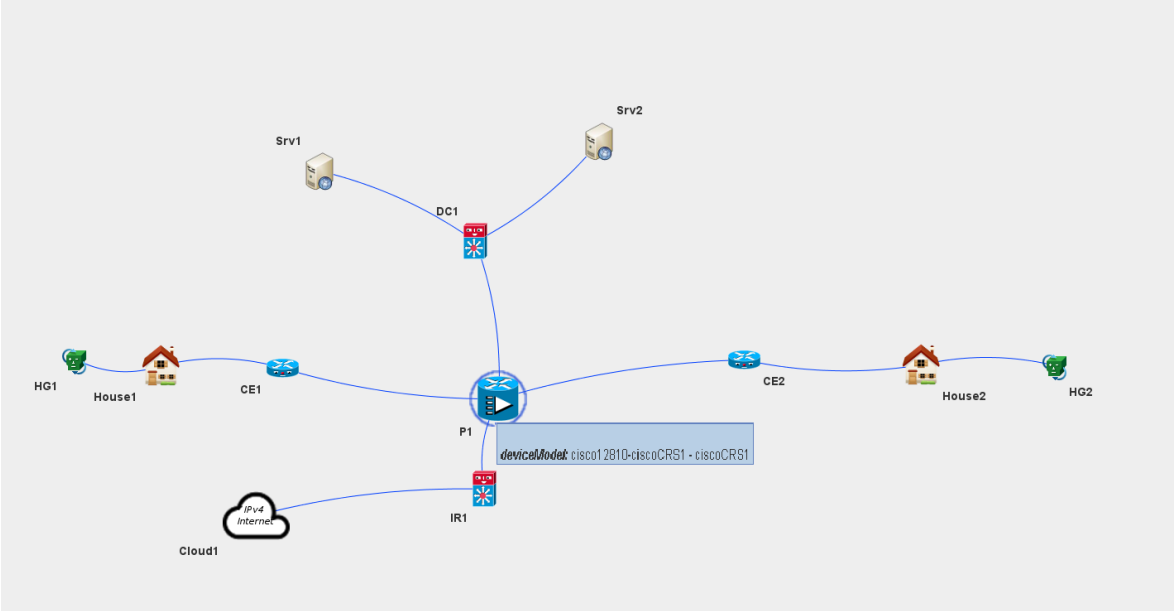
Фигура 5-43 Сравнение между състояния “CE able to translate IPv6 to IPv4” и „Building Automation in Production“



5.9.5 Network IPv6 Capable

Състояние “IPv6 Capable” се достига след подмяна на опорния маршрутизатор P1 с по-нов модел устройство. Процесът на обновяване на текущото състояние отчита промяна в свойствата на устройството, като модела е подменен от cisco12810 на ciscoCRS1. Моделът е извлечен чрез SNMP OID “sysObjectID”.

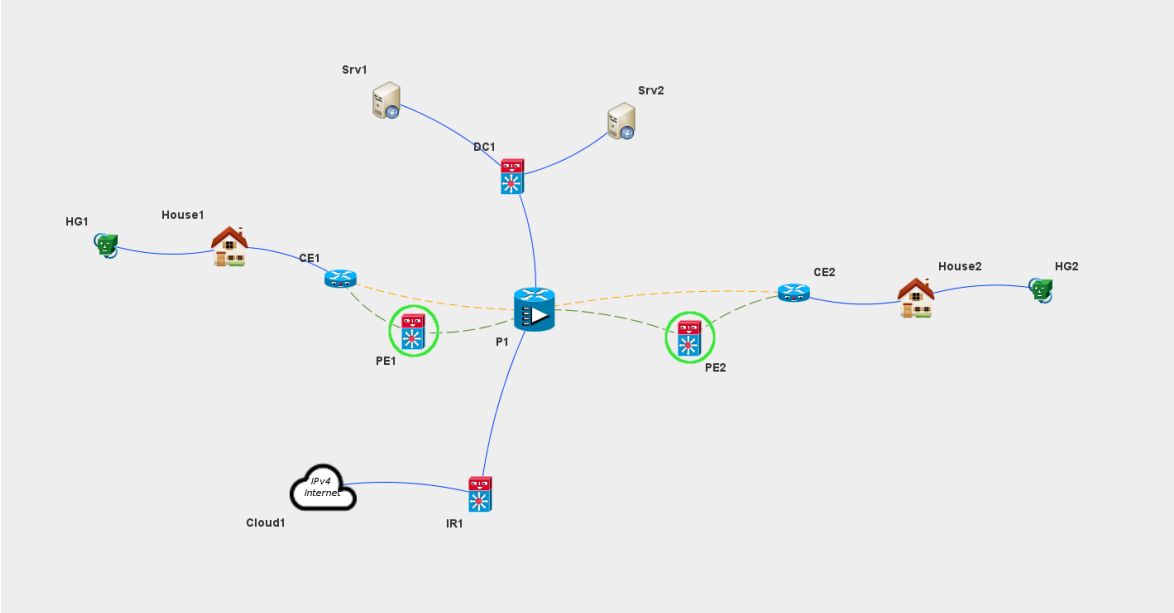
Фигура 5-44 Сравнение между състояния “Network IPv6 Capable” и „Building Automation in Production”



5.9.6 Network extended

Състояние „Network Extended” се постига чрез добавяне на нов слой от PE маршрутизатори между слоя за достъп и слоя на опорната мрежа. Промяната се изразява в добавянето на две нови устройства, отпадане на връзките между P1 и CE1/ CE2 и появата на нови връзки между CE1/2 –PE1/2 и PE1/2 – P1. Промените в мрежовото състояние са открити чрез извличане на SNMP таблици като lldpRemoteSystemsData, cdpCache, ipCidr.

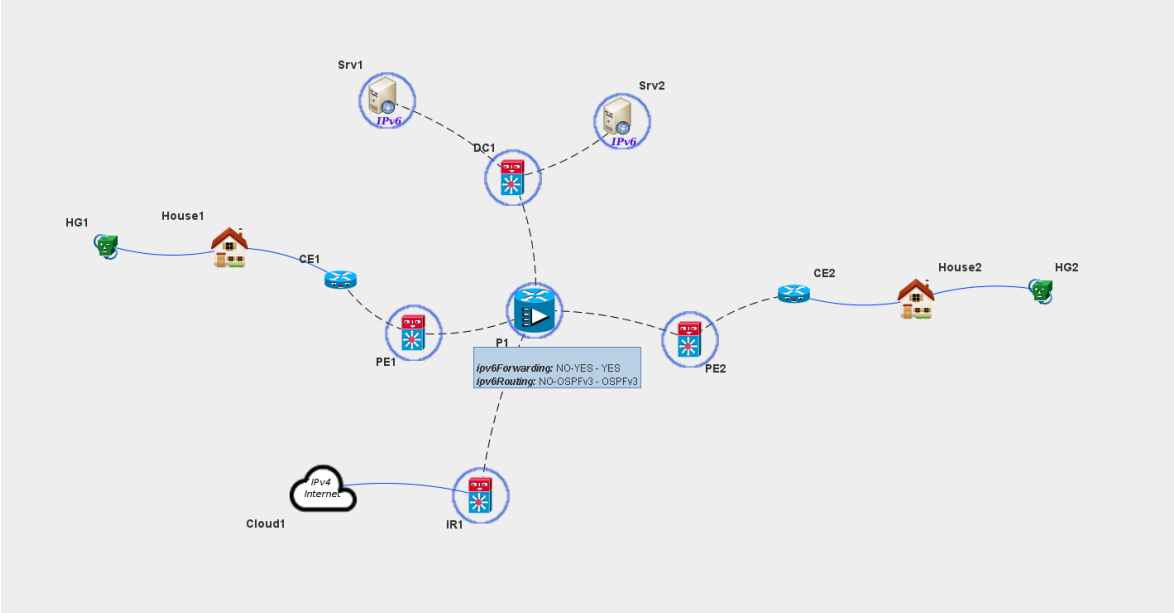
Фигура 5-45 Сравнение между състояния “Network Extended” и „Network IPv6 Capable”



5.9.7 IPv4+IPv6

Състояние “IPv4+IPv6” се характеризира с пълен двоен IP стек в цялата мрежа. За целта е добавен IPv6 на всички устройства без CE1 и CE2, които вече поддържат двоен IP стек. Промените се изразяват в добавяне на нови свойства ipv6forwarding=“YES” и ipv6Routing=”OSPFv3”.

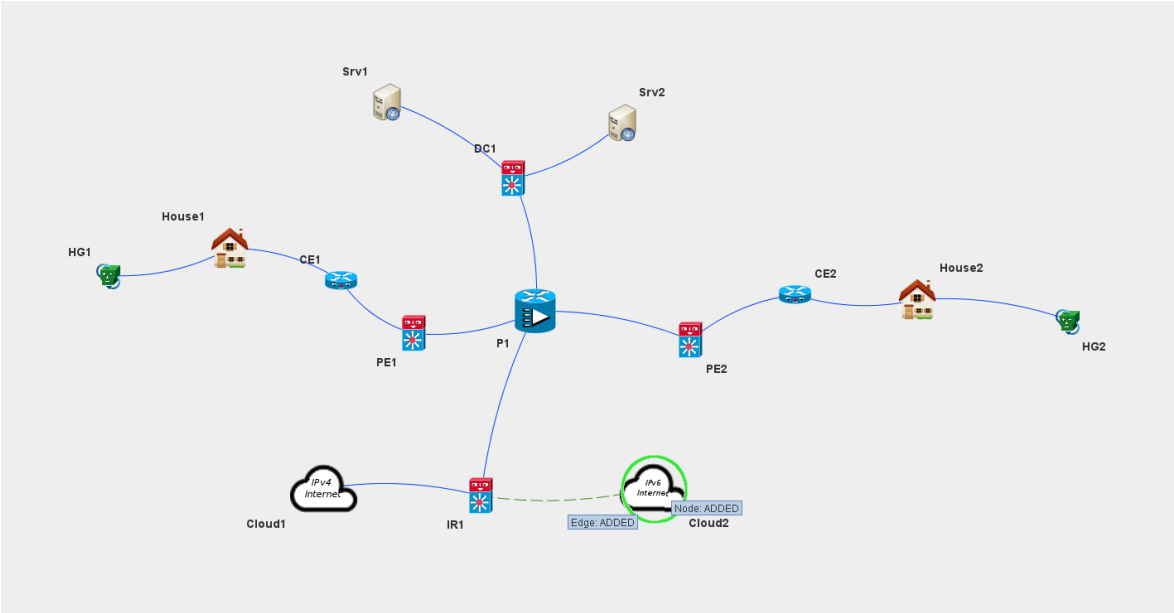
Фигура 5-46 Сравнение между състояния “IPv4+IPv6” и “Network Extended”



5.9.8 Network linked to IPv6 Internet

Свързването на мрежата към вече съществуващия IPv6 Internet е важна стъпка в процеса на еволюционното развитие на мрежовата инфраструктура. В конкретния случай Internet пространството е представено като два отделни облака – IPv4 Internet и IPv6 Internet. Промяната е изразена, като добавяне на нов възел Cloud2 и добавянето на нова връзка между Cloud2 и IR1.

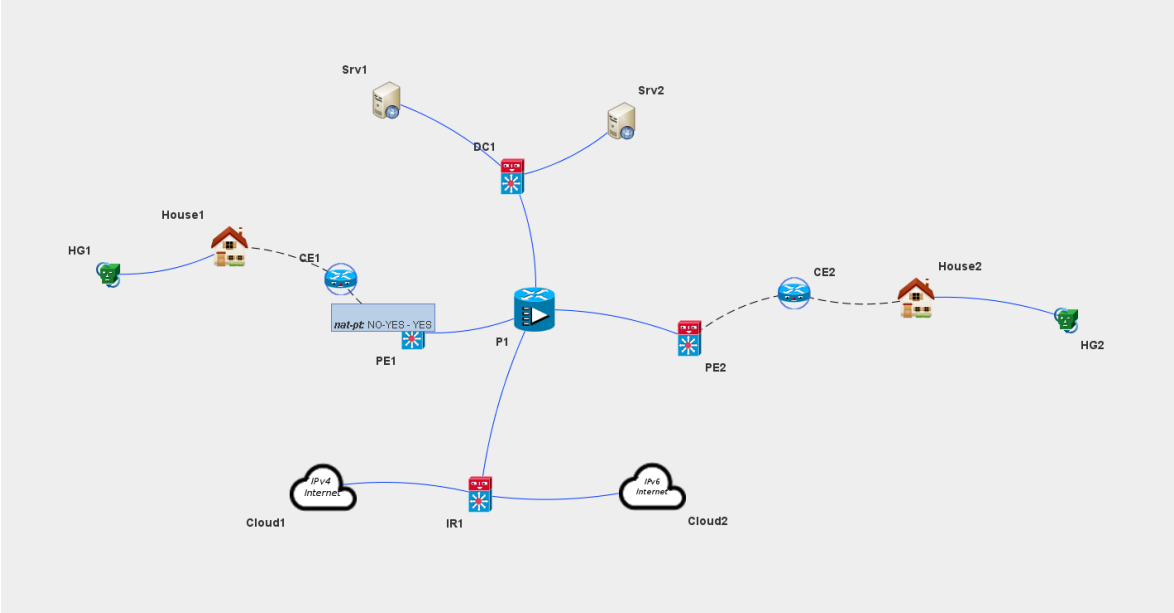
Фигура 5-47 Сравнение между състояния “IPv4+IPv6” и „Network linked to IPv6 Internet”



5.9.9 NAT-PT free network

NAT-PT е състояние, което се достига чрез отстраняване на първоначалния механизъм за превод на адреси от CE1 и CE2. NAT-PT вече е ненужен, тъй като мрежата поддържа изцяло двоен IP стек. Промените се изразяват в промяна на свойствата на възли CE1/ CE2 и на връзките между CE1/2 и съседните им възли.

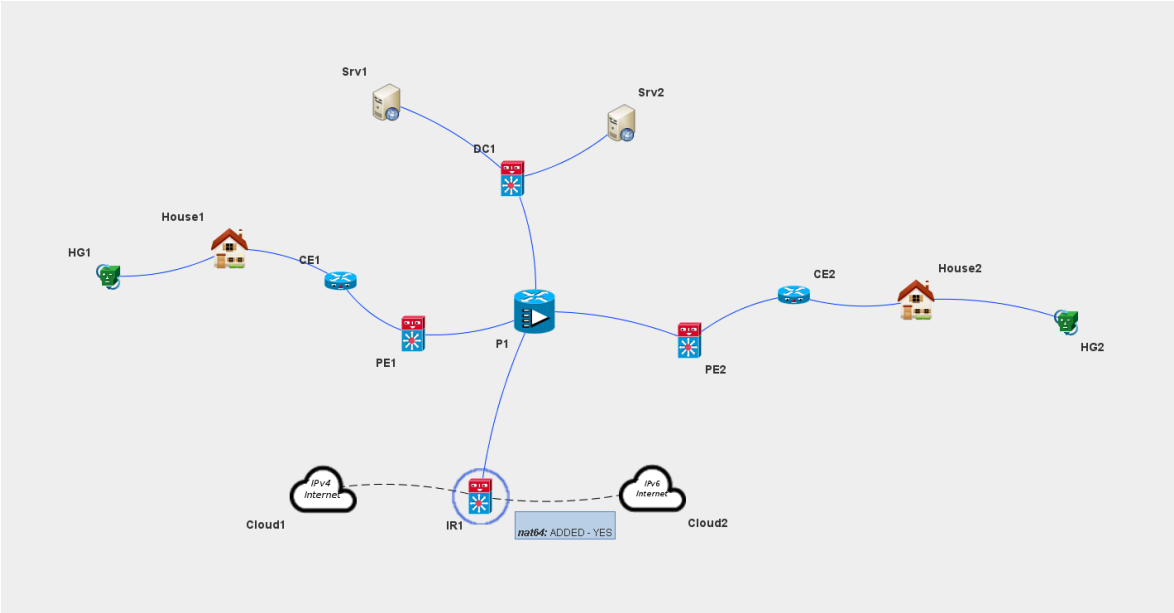
Фигура 5-48 Сравнение между състояния „Network linked to IPv6 Internet" и "NAT-PT free Network"



5.9.10 Network able to translate between IPv4 and IPv6

Състоянието се достига след прилагане на втория механизъм за превод на адреси, част от стратегията. Механизмът NAT64 служи за глобален превод от IPv4 към IPv6 и обратно. Най-подходящото място за приложение на NAT64 е на възел IR1 и връзките му с IPv4/IPv6 Internet.

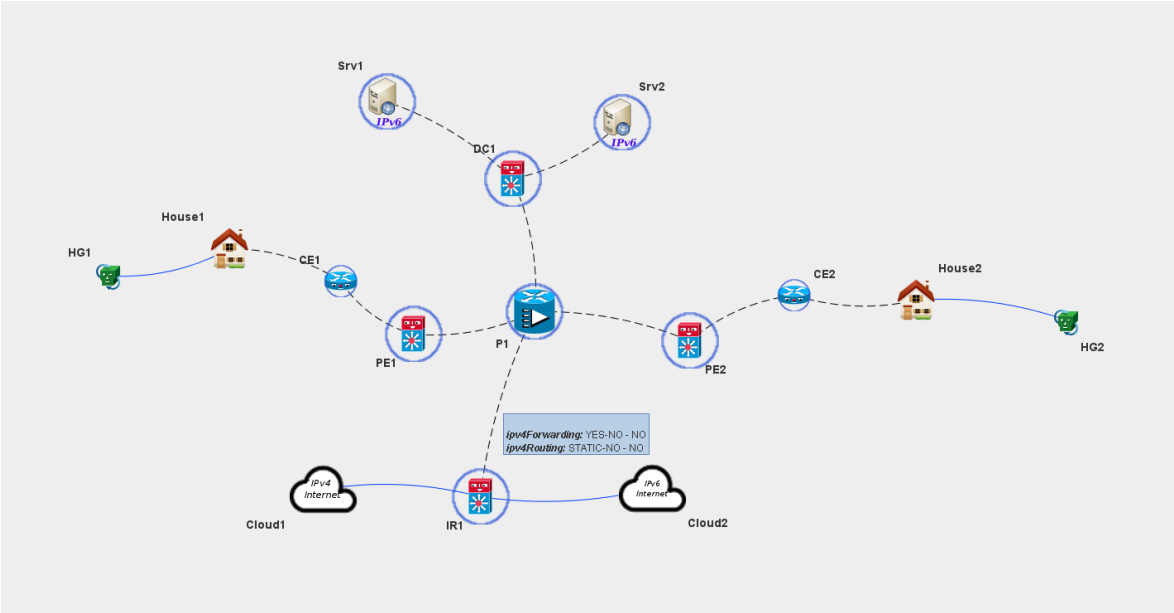
Фигура 5-49 Сравнение между състояния "NAT-PT free Network" и „Network able to translate between IPv4 and IPv6"



5.9.11 IPv4 free network

Състоянието се достига след забраняване на IPv4 на всеки един възел на мрежата, без IR1. Важно е да се отбележи, че стъпката се получава чрез отстраняване на конфигурираните IPv4 адреси от интерфейсите на различните устройства. За момента производителите не поддържат опция, чрез която да се забрани IPv4 на глобално ниво.

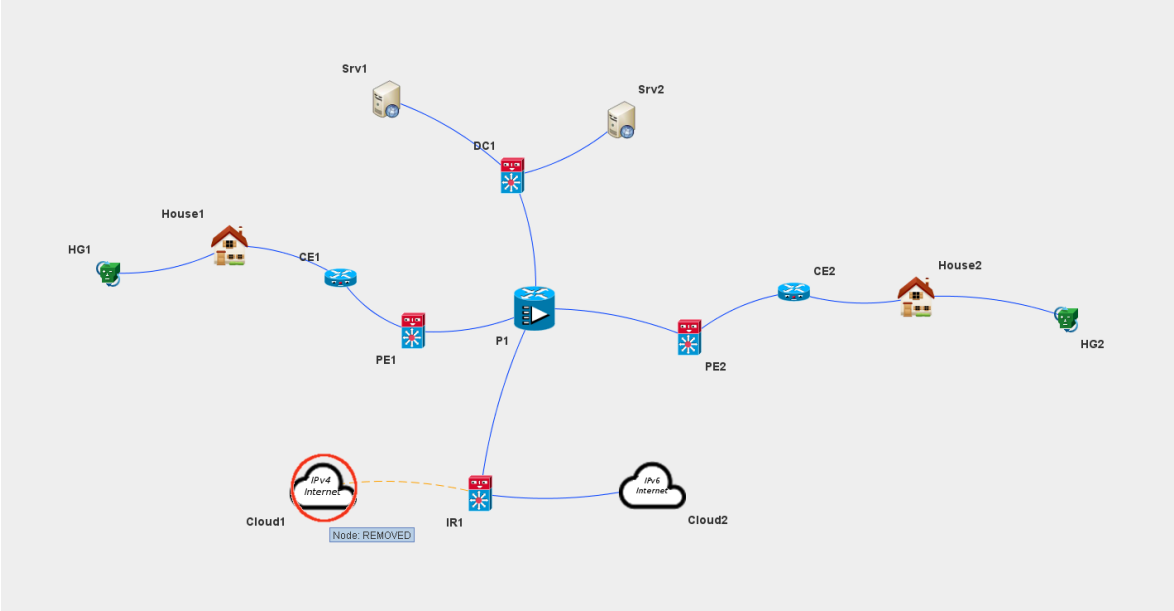
Фигура 5-50 Сравнение между състояния „Network able to translate between IPv4 and IPv6“ и „IPv4 free Network”



5.9.12 IPv6 Only

Това е желаното състояние – мрежа, която е изцяло IPv6 базирана. Достига се чрез отстраняване на NAT64 от IR1 и премахване на вече ненужната връзка с IPv4 Internet.

Фигура 5-51 Сравнение между състояния „IPv4 free Network" и желаното „IPv6 Only"



5.10 Обобщение

В настоящата глава авторът предлага решение на проблема с липсата на софтуерни средства, които да подпомогнат прехода към IPv6. За целта е създаден прототип на софтуерна система, способен да подпомогне мрежовите инженери и архитекти, които ще извършват подобни преходи. Прототипът е способен да разкрие текущото състояние на мрежата, да попълва различни модели от данни с данни за състояние, да се интегрира с други OSS/BSS системи, да покаже разликите между две състояния на мрежата и да изпълни стъпките от стратегията, избрана за еволюционен път. При изпълнението на всяка една стъпка, първо се проверява дали са изпълнени техническите ограничения, след това се изпълнява действието, накрая моделът на мрежата се обновява и се проверява дали е постигнат необходимия ефект.

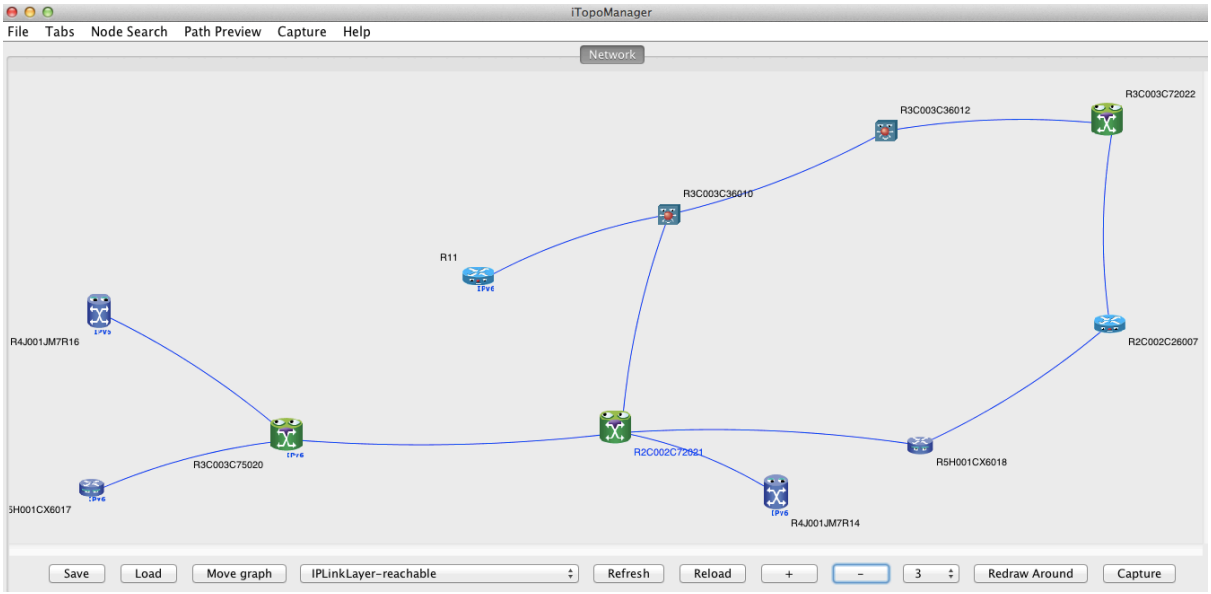
Глава 6: Разработени програмни системи

6.1 Прототип на система за трансформация на мрежи

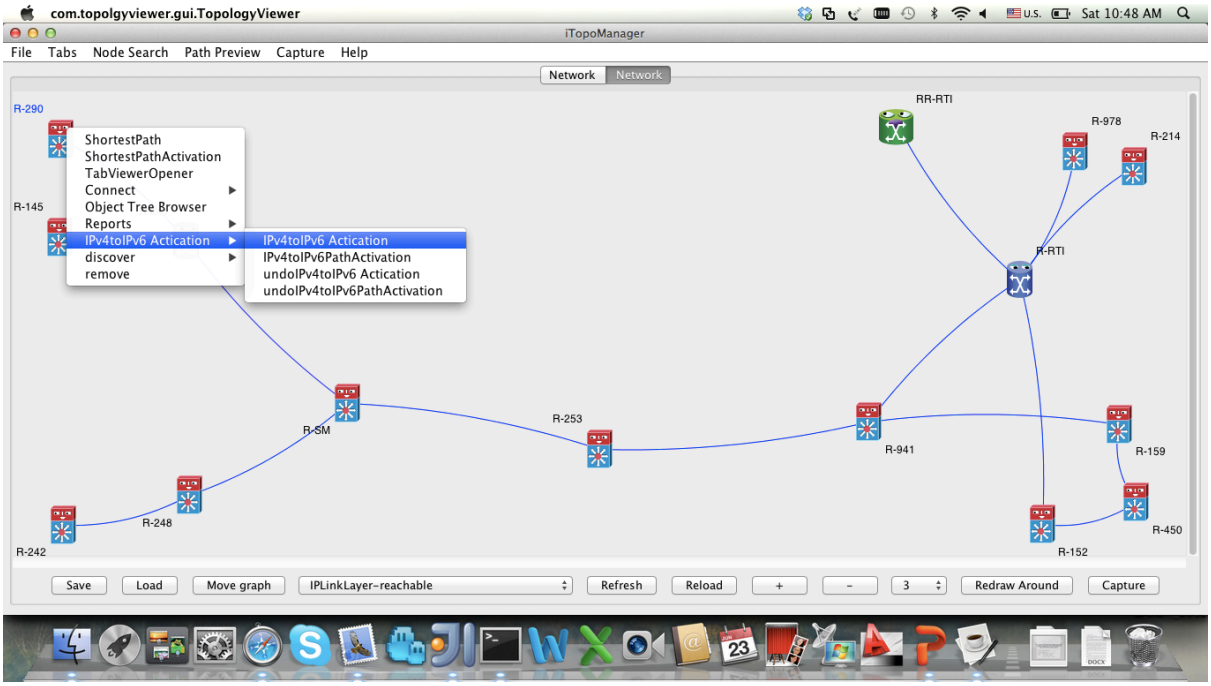
Разработения от автора прототип бе разпространен и приложен в мрежовите инфраструкутри на множество оператори и бизнес организации.

Изгледи от различни реални мрежови топологии са демонстрирани на Фигури 6-1 - 6.4. С цел гарантиране сигурността на отделните организатори, имената и адресите от реалните устройства са подменени с други – автоматично генерирани такива.

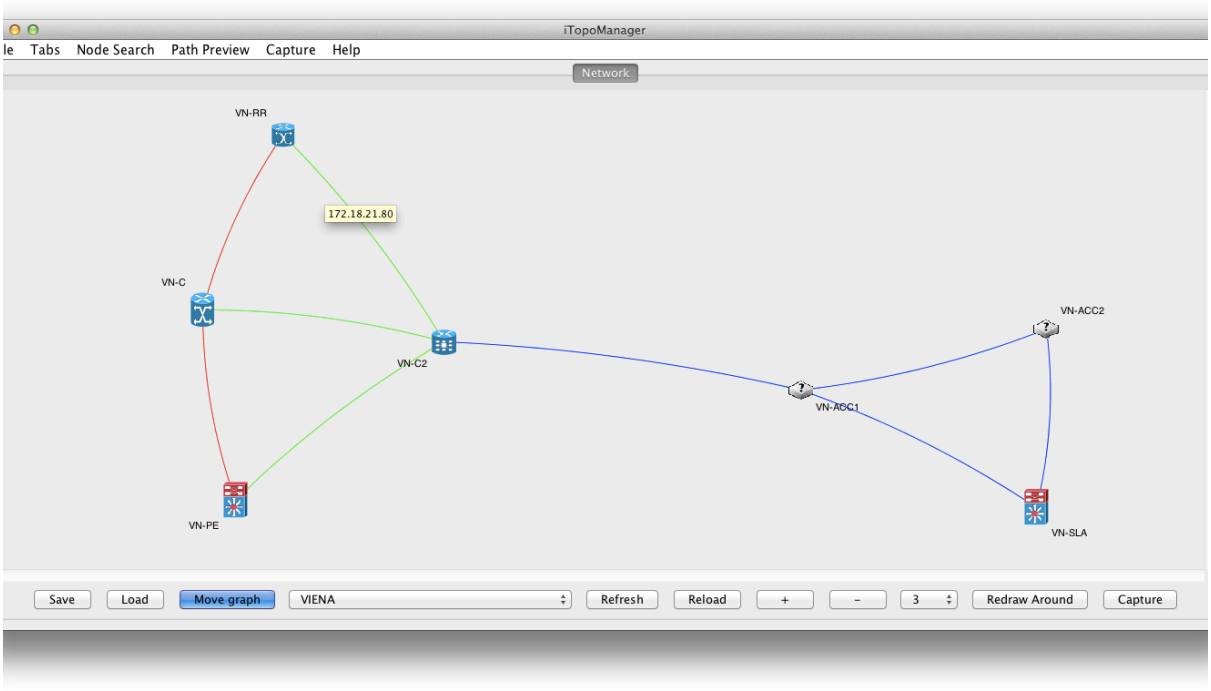
Фигура 6-1 Мрежа с оборудване на Cisco, Huawei, Juniper – филтър по IP свързаност



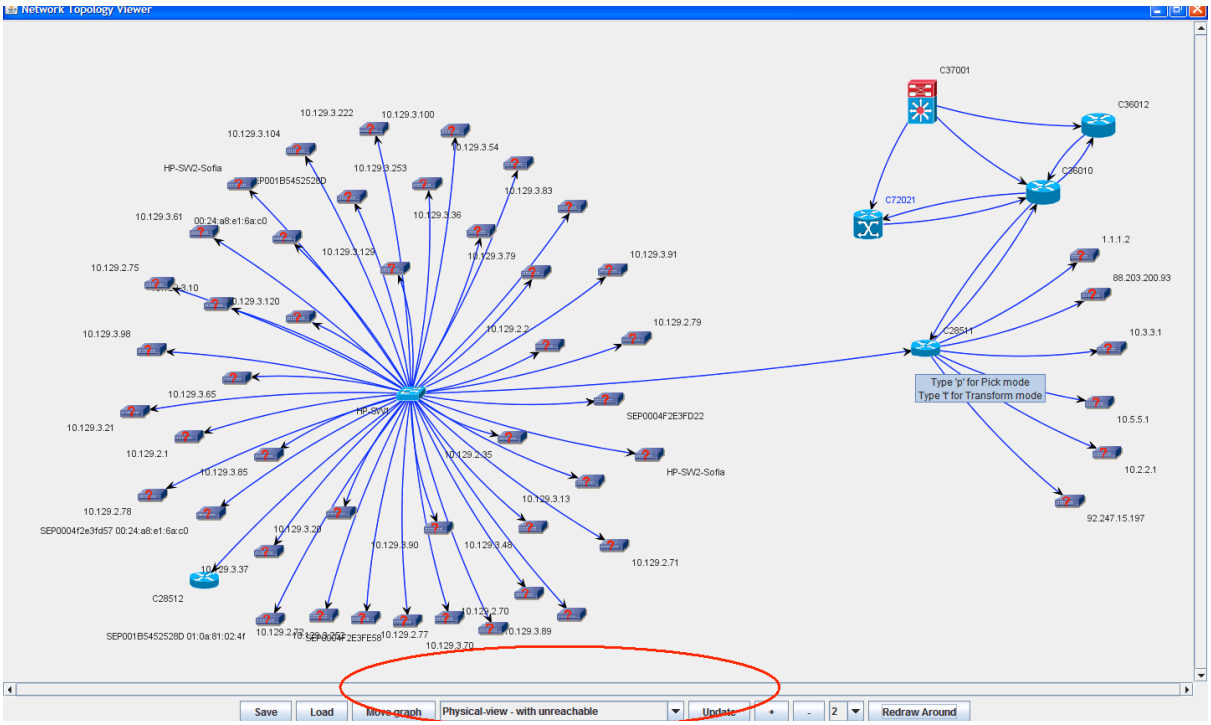
Фигура 6-2 Мрежа съставена от CISCO 76xx, RR-RTI и R-RTI са BGP route-reflectors и са на друг производител на техника (Riverstone)



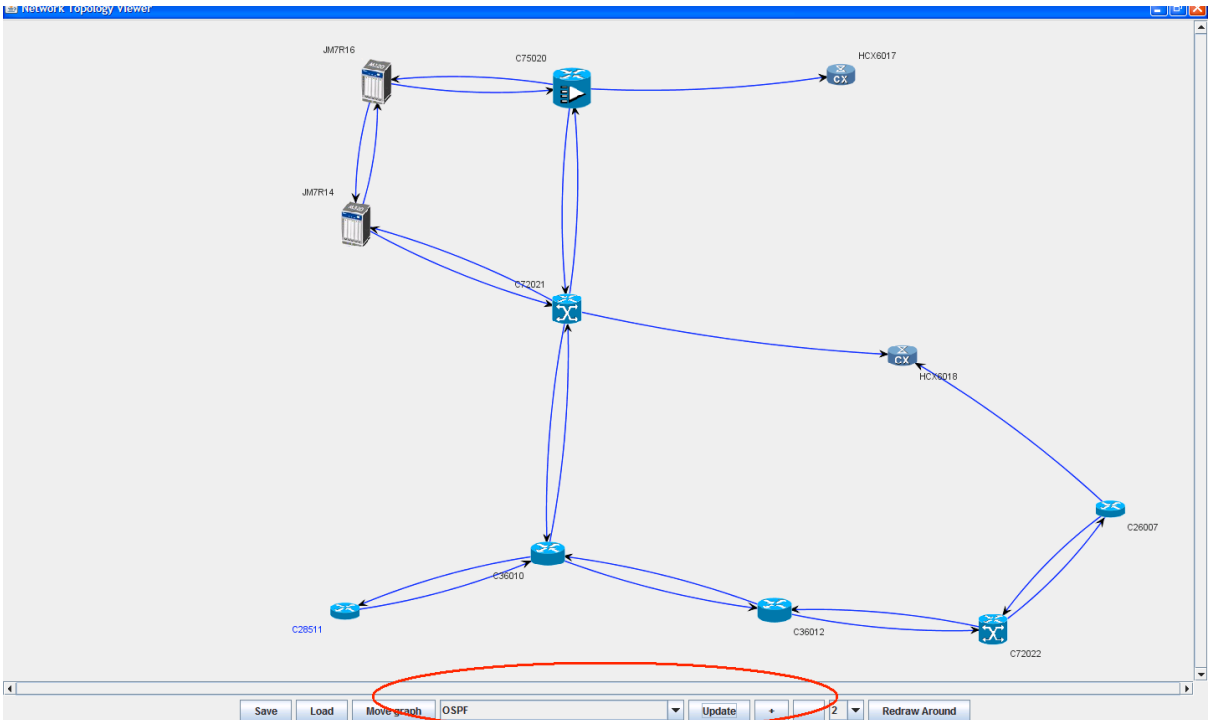
Фигура 6-3 Филтриране по местоположение (показани са устройствата във Виена)



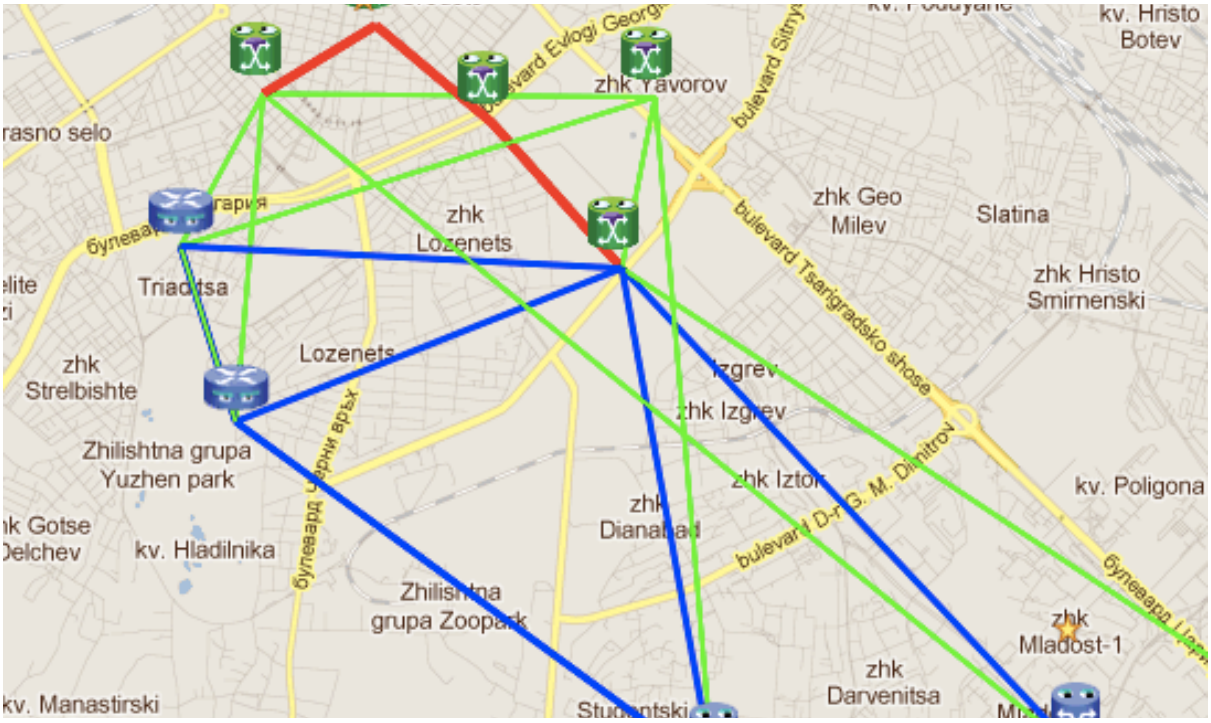
Фигура 6-4 В топологията са включени мрежовите и разкритите крайни устройства (компютри или модеми)



Фигура 6-5 Филтрация по маршрутизиращ протокол - OSPF



Фигура 6-6 Демонстрация на топология върху топографска карта (Google Maps)

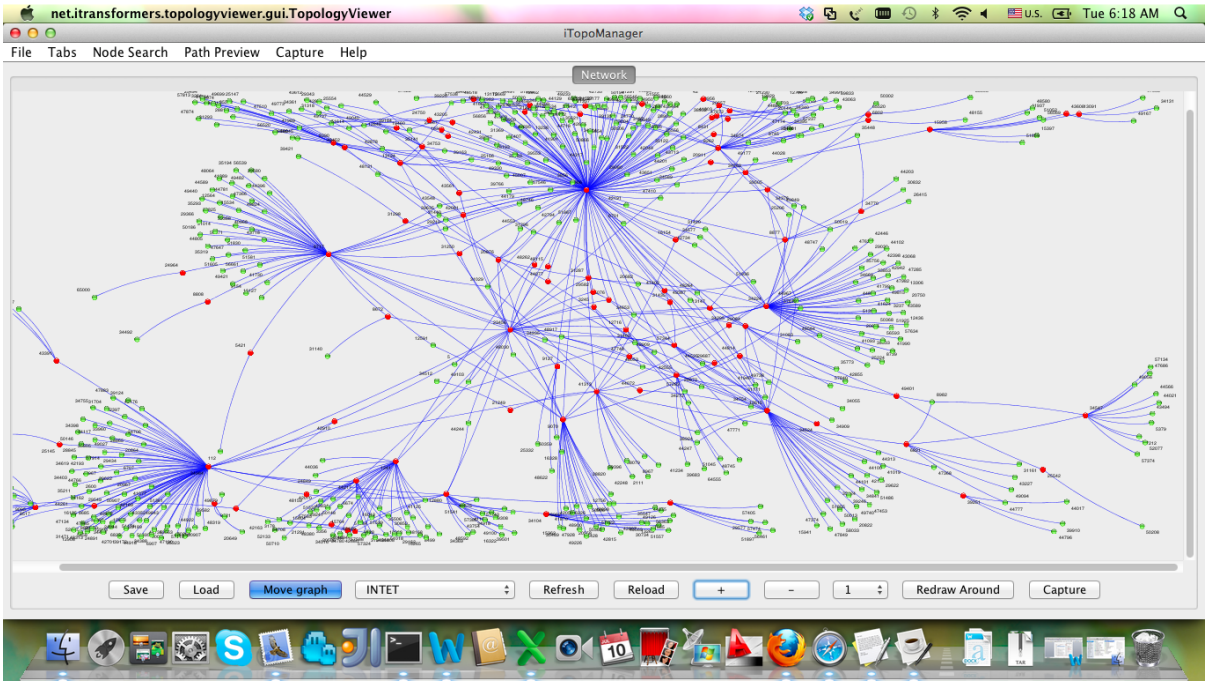


6.2 InternetMap

InternetMap е софтуер, базиран на на основния прототип, целящ да изчертае карта на свързаността между различните BGP автономни системи на различните доставчици в Интернет. Целта на създаването му бе да покаже, че създаденият от Автора механизъм за разкриване на мрежови инфраструктури може да се използва без съществени промени и за генериране на други модели и съответно за изчертаването им.

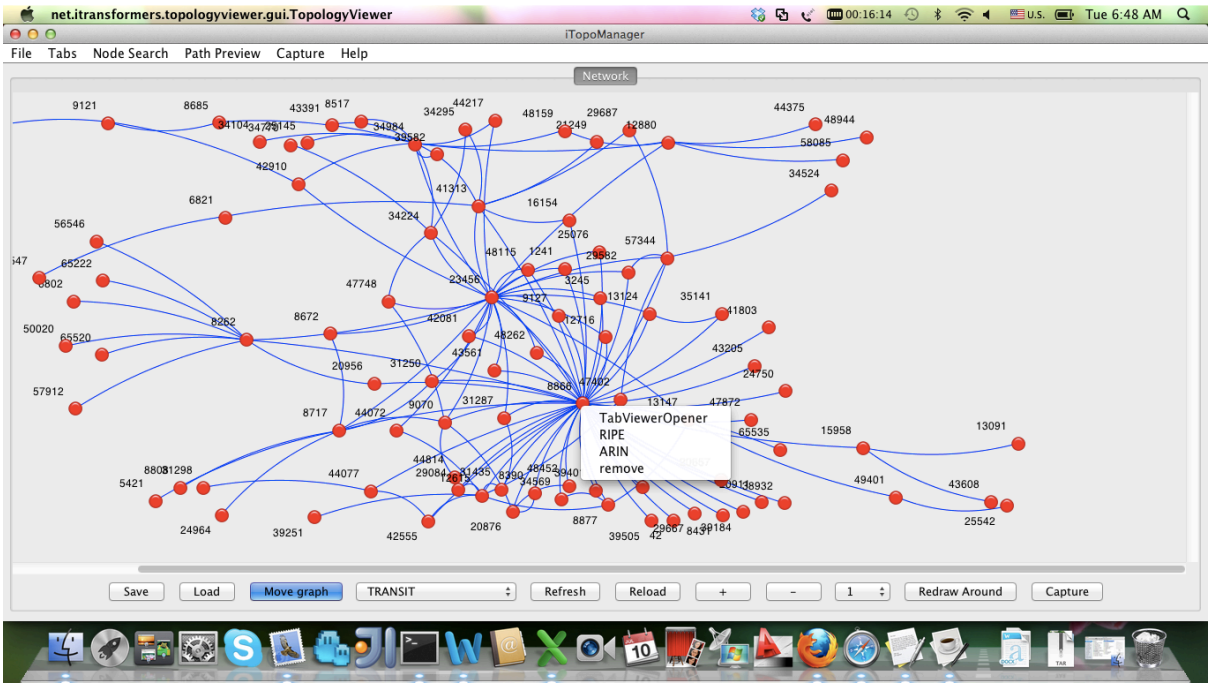
На Фигура 6-7 е демонстрирана карта на Българското Интернет пространство. Червените точки представляват номерата на транзитните автономни системи, а зелените на крайните.

Фигура 6-7 InternetMap (Bulgarian BG peering)



На Фигура 6-8 е демонстрирана същата карта, но е приложен филтър и са оставени единствено транзитните автономни системи. На фигурата е показано и RightClick меню, което има 4 метода. Методите RIPE & ARIN предоставят възможност на потребителя да получи информация за дадената автономна система, за организацията която стои зад нея, за административно отговорните лица и за разпространените от нея IP маршрути в глобалната Интернет таблица.

Фигура 6-8 InternetMap (Bulgarian BG peering – транзитни автономни системи)



Заключение и резюме на получените резултати

В разработения дисертационен труд е отразена литературна справка от български и английски автори и разработки. Резултатите от проучванията показват огромно разнообразие от мрежови технологии и механизми за преход от IPv4 към IPv6. Въпреки това многообразие, преходът не е масова практика, особено при големите доставчици на мрежова свързаност.

Част от проблема с прехода е и наличието на голямо количество софтуер за управление на бизнеса и мрежата, който също трябва да бъде адаптиран към новия протокол.

Проведеният анализ ясно показва необходимостта от разработка на нов подход към прехода от IPv4 към IPv6. Подходът трябва да бъде независим от различните мрежови технологии и да работи еднакво добре с всеки един от механизмите за преход.

В резултат от извършените изследвания и работата по дисертационния труд са получени следните основни резултати от научно-приложен и изцяло приложен характер:

1. Обстойно са изследвани съществуващите мрежови технологии под и над IP слоя, архитектурата на мрежата на съвременен доставчик на услуги, IPv4/v6 протокол,

процеса на раздаване на адреси при IPv4 и съответно при IPv6, и механизмите за преход от IPv4 към IPv6.

2. Анализирана е архитектурата на съществуващите системи за управление на мрежата и бизнеса, като са изследвани NGOSS, SID модела, MTOSI, OSS/J. Идентифицирани са възможности за интеграция на системата за еволюция на мрежата със съществуващия OSS/BSS.
3. Предложен е подход за решение на проблема с прехода от IPv4 към IPv6. Подходът разглежда процеса на миграция като преход от едно текущо към друго желано състояние на мрежата.
4. Преходът ще се извърши чрез изпълнение на множество еволюционни стъпки. Всяка една стъпка се състои от технически и бизнес ограничения, действие и ефект върху мрежата. Стъпките може да бъдат групирани в стратегии. Всяка една стратегия може да бъде оценена по дадени технически и бизнес еволюционни критерии. Предложен е алгоритъм за избор на еволюционния път – стратегията, която отговаря най-добре на зададените еволюционни критерии.
5. Разработени са модели на състоянието на мрежата. Основният моделът е графовиден в graphml формат. Устройствата са представени като възли а връзките като ребра. Всеки един възел и всяка една връзка се характеризира с определени свойства. На базата на елементите и техните свойства мрежата се извършва топологична филтрация и визуализация на топологията на различни нива от OSI модела. Метаданните от моделите се използват като входящи параметри на еволюционните стъпки.
6. Подходът е експериментално приложен върху контекста на оператор X. Условията, в които е поставен операторът X са определени на базата на анализ на ситуацията, в която се намират повечето от съвременните телеком оператори. Дефинирани са първоначално и желано състояние на мрежата. Разработени са множество еволюционни стъпки, съответстващи на различните механизми за преход от IPv4 към IPv6.
7. Предложени са четири стратегии за преход от IPv4 към IPv6. Дефинирани са критерии за избор спрямо контекста на оператор X. Избрана е стратегия, по която да еволюира мрежата на базата на алгоритъма за еволюционния път.

8. Разработен е прототип на системата за еволюция на мрежата. Прототипът е способен да разкрие текущото състояние на мрежата, да попълни модела на състоянието, да се интегрира с останалите OSS/BSS приложения чрез попълване на актуални данни за мрежата в техните SID модели, да визуализира разликите между две състояния на мрежата и да изпълни стъпките от стратегията.
9. Прототипът има приложно и педагогическо значение и е успешно интегриран в множество телекомуникационни оператори и в курсовете по IP телекомуникационни мрежи и MPLS опорни мрежи от програма Телекомуникации (електронни комуникации) на магистърски факултет на Нов Български Университет.
10. Разработеният софтуер е разпространен успешно чрез портала за свободен софтуер sourceforge (http://sourceforge.net/projects/itransformer/) като до момента на завършването на дисертационния труд е бил изтеглен над 1000 пъти от потребители от 61 държави.

Публикации по дисертационния труд

1. Milovanov, N., “Traffic optimization in the modern corporate WAN data network”, “Telecom 2009”, Varna, Bulgaria, pp. 328-334, Oct. 8-9, 2012
2. Milovanov, N. , “Service Fulfillment and Assurance in the NGN Networks”, Journal of NVU, Veliko Tarnovo, Bulgaria, 2009
3. Milovanov, N., From Static to Dynamic QoS”, Годишник на департамент "Телекомуникации", 2008 и 2009, НБУ, София, България
4. Slavinski A., Milovanov N., Georgieva V., “IPv4 TO IPv6 NETWORK TRANSFOMATION”, Годишник на департамент "Телекомуникации", 2008 и 2009, НБУ, София, България
5. Milovanov,N., Slavinski A., Georgieva V., “Service Oriented framework for IPv4 to IPv6 Network transformations”**, “**Infusing Research and Knowledge in South-East Europe”,5th Annual South-East European Doctoral Student Conference, SEERC, Thessaloniki, Greece, pp. 358-370, 2010
6. Milovanov N., Slavinski A., Bogomilov I., “Methodology for analysis and selection of Best Practices in the area of embedded systems and industrial informatics”, “Proceedings of International Conference for Entrepreneurship, Innovation and Regional Development ICEIRD 2011”, Ohrid, Macedonia, pp.1-6, 5-7 May 2011
7. Milovanov.N, Bogomilov I., Slavinski. A, 4TO6TRANS USE CASE - DYNAMIC NETWORK INVENTORY DATA POPULATION“, MANAGEMENT OF TECHNOLOGY - STEP TO SUSTAINABLE PRODUCTION“, Bol, Croatia, 2011
8. Милованов Н., Богомилов И. , “Сравнение между IPv4 и IPv6 виртуални частни IPSEC мрежи“, Списание „Инженерни Науки “, София, България, 09.2011
9. Milovanov N., Bogomilov I., “Case Study - IPv6 based building automation solution integration into an IPv4 Network Service Provider infrastructure“, N.Milovanov, I. Bogomilov, CompSysTech '12 Proceedings of the 13th International Conference on Computer Systems and Technologies, pp. 216-223, 2012
10. Богомилов И., Милованов Н., Славински А., Петров Г., „МЕХАНИЗМИ ЗА ПРЕХОД ОТ IPV4 КЪМ IPV6“, Сборник доклади от Юбилейна научна конференция по повод 10 години от създаването на Национален военен университет „Васил Левски”, гр. Велико Търново, 14-15 юни 2012 г

11. Петров Г., Стефанова Т., Богомилов И., Милованов Н., „Мениджмънт аспекти от прилагането на персонални системи за мониторинг на местоположението – аспекти на сигурността“ , „Мениджмънт в динамично променяща се среда за сигурност“, Велико Търново, 2012, Том 5, стр. 142.

Декларация за оригиналност на резултатите

Декларирам, че настоящата дисертация съдържа оригинални резултати, получени при проведени от мен научни изследвания (с подкрепата и съдействието на научния ми ръководител). Резултатите, които са получени, описани и/или публикувани от други, учени са надлежно и подробно цитирани в библиографията.

Настоящата дисертация не е прилагана за придобиване на научна степен в друго висше училище, университет или научен институт.

Подпис:

Списък на използваните съкращения

3DES – Triple DES

6PE - IPv6 Provider Edge

6VPE - IPv6 Virtual Provider Edge

AAA - Authentication, Authorization and Accounting

ACM - Adaptive Modulation and Coding

ADSL - Asynchronus DSL

AES - Advanced Encryption Standard

AES - Advanced Encryption Standard

AFTR - Address Family Translation Router

AH – Authentication header

AJAX - Asynchronous JavaScript and XML

AMR - Adaptive Multiple Rate

AP - Access Points

APON – ATM PON

ARIN - American Registry for Internet Numbers

ARP - Address Resolution Protocol

ARPA - Advanced Research Projects Agency

ASN - Access Service Network

ATIS - Alliance for Telecommunications Industry Solutions

ATM – Asynchronus Transfer Mode

AuC - Authentication Center

BGP – Border Gateway Protocol

BGP (Border Gateway Protocol)

BPON – Broadband PON

BR - Border Router

BSC - Base Station Controller

BTS - Base Transceiver Station

CAMEL - Customised Applications for Mobile Enhanced Logic

CAP - Carrierless Amplitude Phase

CAP - Carrierless Amplitude Phase

CE - Customer Edge

CERN - *Conseil Européen pour la Recherche Nucléaire*

CG - Customer Gateway

CJK - China Japan and Korea

CLI – Command Line Interface

CN – Core Network

CNAME- Canonical Name

CO – Central Office

CPE – Customer Premises Equipment

CPON – CDMA PON

CRC - Cyclic Redundancy Check

CS - Circuit Switched

CSMA/CD - Carrier Sense Multiple Access with Collision Detection

CSN - Connectivity Service Network

DARPA- Defence Advanced Research Projects Agency

DC - Data Center Router

DDOS - Distributed Denial Of Service

DES - Data Encryption Standard

DHCP - Dynamic Host Configuration Protocol

DMO - Direct Mode Operation

DMT- Discrete Multi-Tone

DNS - Domain Name Service

DOCSIS - Data Over Cable System Interface Specification

DS – Dual Stack

DSL- Digital Subscriber Line

E-DCH - Enhanced Dedicated Channel

EGP (Exterior Routing Protocol)

EIR - Equipment Identity Register

eNodeB - evolved NodeB

EPC - Evolved Packet Core

EPON – Ethernet PON

EPS - Evolved Packet System

ESP - Encrypted Security Payload

ETSI-TISPAN - European Telecommunication Standards

EUI-64 - Extended Unique Identifier

E-UTRAN - Evolved UTRAN

FEC – Forward Error Correction

FQDN - Fully Qualified Domain Name

GEM – GPON Encapsulation Method

GERAN - GPRS/ EDGE radio Access Network

GEPON – Gigabit Ethernet PON

GPON – Gigabit PON

GSM - Global System for Mobile Communications

GTC – GPON Transmission Convergence

HARQ - Hybrid Automatic Repeat reQuest

HDSL - HightSpeed DSL

HG - Home Gateway

HLR - Home Location Register

HSDPA - High-Speed Data Packet Access

HS-DSCH канал - High Speed Downlink Shared Channel

HSS - Home Subscriber Server

HSUPA - High-Speed Uplink Packet Access

HTTP - Hypertext Transfer Protocol

IAB - Internet Advisory Board

ICANN - Internet Corporation for Assigned Names and Numbers

ICCB - Internet Configuration Control Board

ICMP - Internet Control Message Protocol

IDEA - International Data Encryption Algorithm

IEEE- Electrical and Electronics Engineers

IEC - International Electrotechnical Commission

IETF - Internet Engineering task Force

IGP(Interior Gateway Protocol)

IMEI - International Mobile Equipment Identity

IMPs - Interface Message Processors

IMS - IP Multimedia Subsystem

IoT - Internet of Things

IPSEC – Internet Protocol Security

IPTO - Information Processing Techniques Office

IR - Internet Router

ISAKMP - Internet Security Association and Key Management Protocol

ISATAP - Intra-Site Automatic Tunnel Addressing Protocol

ISDN - Integrated Services Digital Network

ISIS - Intermediate System to Intermediate System

ISO - International Organization for Standardization

ITU - International Telecommunications Union

LLC - Logical Link Control

LLID – Logical Link Identifier

LLQ – Low Latency Queuing

LSN –Large Scale NAT

LTE - Long Term Evolution

MAC – Media Access Control

MBMS - Multimedia Broadcast and Multicast Service

MELPe - Mixed Excitation Liner Predictive

MGCP - Media Gateway Control Protocol

MGW - Media Gateway

MIMO - Multiple Input Multiple Output

MME - Mobility Management Entity

MP-BGP – Multi Protocol Border Gateway Protocol

MPLS - Multiprotocol Label Switching

MS - Mobile Station

MSC - Mobile Services Switching Center

MSCS - MSC server

NAT – Network Address Translation

NAT-PT - Network Address Translation/Protocol Translation

NBMA - Non Broadcast Multiple Access

NDP - Neighbor Discovery Protocol

NETCONF – Network Configuration Protocol

NGN - Next Generation Network

NGN-GSI - Next Generation Network – Global Standards Initiative

NNI – Network to Network Interface

NRZ – Non - Return to Zero

NSP - Network Service Provide

OAN – Optical Access Network

ODN – Optical Distribution Network

OFDM - Orthogonal Frequency Division Multiplexing

OFDMA - Orthogonal Frequency Division Multiplexing and Multiple Access

OLT – Optical Line Terminal

ONU – Optical Network Unit

OSI - Open Systems Interconnection

OSPF - Open Shortest Path First

OSPF (Open Shortest Path First)

P - Provider

PABX - Private Automatic Branch Exchange

PAT – Port Address Translation

PCBd – Physical Control Block downstream

PCRF - Policy Control and Charging Rules Function

PDN – Packet Data Network

PE - Provider Edge

P-GW - PDN Gateway

PLI – packet Length Identifier

PLOAM – Physical Layer Operations, Administration and Maintenance

PLOAMd – Physical Layer Operations, Administration and Maintenance downstream

PLOAMu– Physical Layer Operations, Administration and Maintenance upstream

PON – Passive Optical Networks

PPP - Point to Point Protocol

PS – Packet Switched

PSTN - Public Switched Telephone Network

PT –Port Translation

QAM - Quadrature Amplitude Modulation

QCI - QoS Class Identifier

QoS - Quality of Service

QPSK - Quadrature Phase Shift Keying

RADIUS- Remote Authentication Dial In User Service

RDISC - Router Discovery

RED – Random Early Detection

RFC - Request For Comment

RFQ - Request for Quotation

RIP – Routing Information Protocol

RNC - Radio Network Controller

RRM - Radio Resource Management

RSS - Really Simple Syndication

RTP - Real-time Transport Protocol

SAE - System Architecture Evolution

SATNET

SDH - Synchronous Digital Hierarchy

SDSL - Symetrical DSL

S-GW - Serving GW

S-GW - Serving GW

SHDSL- Single-pair high-speed DSL

SIM - Subscriber Identity Module

SIP - Session Initiation Protocol

SIIT - Stateless IP, ICMP NAT64 Translation

SLAAC - Stateless Address Auto-Configuration

SNMP – Simple Network Management Protocol

Srv - Server

SS - Subscriber Stations

SSH - Secure Shell

SwMI - Switching and Maintenance Infrastructure

TBS - TETRA Base Station

TCP/IP - Transmission Control Protocol/Internet Protocol

T-CONT – Transmission container

TDD – Time Division Duplex

TDMA - Time Division Multiple Access

TEDS - TETRA Enhanced Data Service

TETRA - Terrestrial Trunked Radio

TFT - Traffic Flow Templates

TLS - Transport Layer Security

TMO - Trunked-Mode Operation

TRAU - Transcoding Rate and Adaptation Unit

UDP- User Datagram Protocol

UE - User Equipment

UMTS - Universal Mobile Telecommunications System

UNI - User Network Interface

UTP - Unshielded Twisted Pair

UTRAN - UMTS Radio Access Network

VC – Virtual Container

VDSL- Very-high-bit-rate DSL

VLR - Virtual Location Register

VLSM - Variable Length Subnet Mask

VOIP - Voice over IP

VP – Virtual Path

VPN - Virtual Private Network

VRF - Virtual Routing and Forwarding

WDM - Wave Division Multiplexing

WiMAX - Worldwide Interoperability for Microwave Access

WRED – Weighted Random Early Detection

WPON – WDM PON

WWW - World Wide Web

XPATH - XML Path Language

Библиография

1. **Zimmerman H.** *OSI reference model -- The ISO model of architecture for open systems interconnection.* s.l. : IEEE Transactions on Communications, 1980. pp. 425-432. Vol. 28.

2. **Deering, S. and Hinden, R.** *IETF RFC 1883 Internet Protocol, Version 6 (IPv6) Specification.* December 1995.

3. slashdot. [Online] 2013. http://tech.slashdot.org/story/13/01/04/199253/worldwide-ipv6-adoption-where-do-we-stand-today.

4. **Bernstein D. J.** The IPv6 mess. [Online] http://cr.yp.to/djbdns/ipv6mess.html.

5. **Yadav A., Pushkar A.,.** IPv6 protocol adoption in the U.S.: Why is it so slow? [Online] 2012. http://morse.colorado.edu/~tlen5710/12s/IPv6Protocol.pdf.

6. **Atlas A., Nadeau T., Ward D.,.** *draft-atlas-i2rs-problem-statement-00.* s.l. : IETF, Network Working Group, 2013.

7. *IPv4 to IPv6 Transformation Schemas.* **Miyakawa Sh.** 5, s.l. : IEICE TRANC. COMMUN., 2010, Vols. E93—B.

8. *An examination of IPv4 and IPv6 Networks: Constraints and Various Transition mechanisms.* **Govil, J., Kaur, N.,.** s.l. : Southeastcon, IEEE, 2008. 978-1-4244-1884-8/08.

9. *Transition From IPv4 To IPv6: The Method for Large Enterprise Networks.* **Nguyen Ph., Nguyen Qun., Utriainen J.,.** s.l. : IARIA, 2012. INNOV 2012, The First International Conference on Communications, Computation, Networks and Technologies. pp. 5-14. 978-1-61208-244-8.

10. *Case Study - IPv6 based building automation solution integration into an IPv4 Network Service Provider infrastructure.* **Milovanov N., Bogomilov I.,.** RUSE : ACM, 2012. pp. 216-223.

11. **Griffin S.** *Internet Pioneers.* [Online] http://www.ibiblio.org/pioneers/.

12. **Baran P.** *On distributed communications.* s.l. : RAND, 1964. RM-3420-PR.

13. *The World's First Web Published Book.* [Online] http://www.livinginternet.com/i/i.htm.

14. **Berners-Lee T., Cailliau R., Groff J.-F.** *World Wide Web. Flyer distributed at the 3rd Joint European Networking Conference.* Innsbruck,Austria : s.n., 1992.

15. The Victorian Internet. [Online] http://en.wikipedia.org/wiki/The_Victorian_Internet .

16. **DARPA.** *DoD Standard Internet Protocol.* January 1980. IETF RFC 760.

17. **Postel, J.** *IETF RFC 791 Internet Protocol.* 1981.

18. 6INIT. [Online] http://www.6init.org/.

19. Euro6IX. [Online] http://www.euro6ix.org/main/index.php.

20. 6NET. [Online] http://www.6net.org/.

21. 6DEPLOY. [Online] http://www.6deploy.org/.

22. **Deering, S. and Hinden, R.** *IETF RFC 2460 Internet Protocol, Version 6 (IPv6) Specification.* 1998.

23. **Deering , S. and Hinden, R.** *IETF RFC 3414 Internet Protocol Version 6 (IPv6) Addressing Architecture.* 2003.

24. **Mockapetris P.** *RFC 1035 - DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION.*

25. **Crawford M., Huitema C.,.** *RFC 2874 - DNS Extensions to Support IPv6 Address Aggregation and Renumbering.* 2000.

26. **Bagnulo, M., Matthews, P. and van Beijnum, I.** *IETF RFC 6146 Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers.* 2011.

27. **Hinden, R., Deering, S. and Nordmark, E.** *IETF RFC 3587 IPv6 Global Unicast Address Format.* August 2003.

28. **Narten, T. and Draves, R.** *IETF RFC 3041 Privacy Extensions for Stateless Address Autoconfiguration in IPv6.* January 2001.

29. **Narten, T., et al.** *IETF RFC 4861 Neighbor Discovery for IP version 6 (IPv6).* 2007.

30. **Conta A., Deering S.** *RFC 4443 - Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification.* 2006.

31. **Thomson S., Narten T.,.** *RFC 4862 - IPv6 Stateless Address Autoconfiguration.* 2007.

32. **Droms R., Bound J., Volz B.,.** *RFC 3315 - Dynamic Host Configuration Protocol for IPv6 (DHCPv6).* s.l. : IETF, 2003.

33. **Simpson W.** *RFC 1661 - The Point-to-Point Protocol (PPP).* s.l. : IETF, 1994.

34. **Draves, R.** *IETF RFC 3484 Default Address Selection for Internet Protocol version 6 (IPv6).* February 2003.

35. **Gilligan, R. and E. Nordmark,.** *IETF RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers.* August 2000.

36. *IMS-centric Evaluation of IPv4/IPv6 Transition Methods in 3G UMTS Systems.* **Bokor L., Kanizsai Z.,.** 3&4, s.l. : IARIA, 2010, International Journal on Advances in Networks and Services,, Vol. 3, pp. 402-416.

37. **Nordmark, E. and Gilligan, R.** *IETF RFC 4213 Basic Transition Mechanisms for IPv6 Hosts and Routers.* 2005.

38. **Durand A., Droms R.,.** *RFC 6333 Dual-Stack Lite.* s.l. : IETF, 2011.

39. **Tsirtsis, G. and Srisuresh, P.** *IETF RFC 2766 Network Address Translation - Protocol Translation (NAT-PT).* February 2000.

40. **Jiang S., Guo D., Carpenter B.,.** *RFC 6264 - An Incremental Carrier-Grade NAT (CGN) for IPv6 Transition.* s.l. : IETF, 2011.

41. **Rekhter, Y., et al.** *IETF RFC 1918 Address Allocation for Private Internet.* 1996.

42. **Aoun, C. and Davies, E.** *IETF RFC 4966 Reasons to Move the Network Address Translator - Protocol Translator.* July 2007.

43. **Li, X., Bao, C. and Baker, F.** *IETF RFC 6145 IP/ICMP Translation Algorithm.* 2011.

44. **Gilligan, R. and Nordmark, E.** *IETF RFC 2893 Transition Mechanisms for IPv6 Hosts and Routers.* 2000.

45. **Hain, T.** *IETF RFC 2993 Architectural Implications of NAT.* November 2000.

46. **Savola, P. and Patel, C.** *IETF RFC 3964 Security Considerations for 6to4.* 2004.

47. **Despres, R.** *IETF RFC 5569 IPv6 Rapid Deployment on IPv4 Infrastructures (6rd).* 2010.

48. **Carpenter, B. and Jung, C.** *IETF RFC 2529 Transmission of IPv6 over IPv4 Domains without Explicit Tunnels.* 1999.

49. **Wu, J., et al.** *IETF RFC 5747 4over6 Transit Solution Using IP Encapsulation and MP-BGP Extensions.* 2010.

50. **Templin, F., Gleeson, T. and Thaler, D.** *IETF RFC 5214 Intra-Site Automatic Tunnel Addressing Protocol (ISATAP).* 2008.

51. **Huitema, C.** *IETF RFC 4380 Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs).* 2006.

52. **Clercq J. De, Ooms D.,.** *RFC 4798 - Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE).* s.l. : IETF, 2007.

53. **Clercq J. De, Ooms D., Carugi M.,.** *RFC 4539 - BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN.* s.l. : IETF, 2006.

54. *Evaluating IPv4 to IPv6 transition mechanisms.* **Raciu, I.** s.l. : IEEE, 2003. Telecommunications, 2003. ICT 2003. 10th International. Vol. 2, pp. 1091 - 1098.

55. **Henrique S., Brito B.** *Brazil, Analysis of Connectivity Level of IPv4 Internet vs Next Generation IPv6 "Island" in.* 2012.

56. **Rexford, J., Dovrolis C.,.** Point/counterpoint Future Internet Architecture: Clean-State Versus Evolutionary Research. *CommuniCations of the ACM.* 2010, Vol. 52, 9.

57. **Garlan D., Shaw M.** *An introduction to software architecture.* s.l. : World Scientific Pub Co, 1993. Vol. II.

58. *The Multi-stakeholder model in the management of Internet critical resources* . **Parra R.** s.l. : ICANN, 2012, Latin America 2012.

59. *Transforming Telecom Management – facing the challenge of next generation networks.* s.l. : Ericsson, January 2009.

60. **Clemm, Al.** *Network Management Fundamentals.* s.l. : Cisco Press, 2006.

61. Telecom OSS/BSS: An overview. [Online] http://ravisharda.blogspot.com/2010/03/telecom-ossbss-overview.html.

62. *ITU M.3010 Principles for a telecommunications management network.*

63. *ITU M.3410 Guidelines and requirements for security management systems to support telecommunications management.*

64. *ITU M.3010-0 eTOM – Introduction.*

65. Introduction to eTOM . [Online] http://www.cisco.com/en/US/technologies/collateral/tk869/tk769/white_paper_c11-541448.html.

66. *ITU M.3050-0 Enhanced Telecom Operations Map (eTOM) Introduction.* 2007.

67. *ITU M.3050-1 eTOM – The business process framework.*

68. *ITU M.3050-2- Process decompositions and descriptions.*

69. *ITU M.3190 Shared information and data model (SID).*

70. *TM Forum GB929 Application Framework (TAM), The BSS/OSS Systems Landscape v.4.3.*

71. *JSR 144: OSS Common API.*

72. *JSR 142: OSS Inventory API.*

73. *JSR 254: OSS Discovery API.*

74. *JSR 263: Fault Management API.*

75. *JSR 89: OSS Service Activation API.*

76. *JSR 91: OSS Trouble Ticket API.*

77. *TM Forum MTOSI 2.1 Release Notes RN306, Approved Version 2.4.*

78. **Sardjana D.** Shareholder Role Analysis of Grameen Telecom in Determining Management Strategy. [Online] 2008. http://www.slideshare.net/djadja/grameen-telecom-stakeholder-thesis-summary.

79. **Ashton K.** That 'Internet of Things' Thing. *RFID Journal.* [Online] 2009. http://www.rfidjournal.com/articles/view?4986.

80. *A Break in the Clouds: Towards a Cloud Definition.* **Vaquero Luis M., Rodero-Merino L.,.** 1, 2009, ACM SIGCOMM Computer Communication Review , Vol. 39 . 50-59.

81. *The GraphML File Format.* [Online] http://graphml.graphdrawing.org/.

82. *EWD 472: Guarded commands, non-determinacy and formal.* [Online] http://www.cs.utexas.edu/users/EWD/ewd04xx/EWD472.PDF.

83. *W3C Recommendation XML Path Language (XPath).* [Online] November 19, 1999. http://www.w3.org/TR/xpath/.

84. **Ф., Андонов.** *АВТОРЕФЕРАТ НА ДИСЕРТАЦИЯ "МЕТОДИ ЗА ГРУПОВО РЕШАВАНЕ НА ЗАДАЧИ НА МНОГОКРИТЕРИАЛНИЯ АНАЛИЗ"* . София : Институт по информационни и комуникационни технологии, 2012.

85. *A New Approach to NGN Evaluation Integrating Simulation and Testbed Methodology* . **Fernandez, M.** 2012. ICN 2012 : The Eleventh International Conference on Networks. 978-1-61208-183-0.

86. *http://cisco.com.* [Online]

87. Cisco IOS. [Online] http://en.wikipedia.org/wiki/Cisco_IOS.

88. *Cisco IOS XR Software.* [Online] http://www.cisco.com/en/US/products/ps5845/index.html.

89. **Krasner G. E., Pop S. T.** *A cookbook for using the Model-View-Controller user interface paradigm in Smalltalk-80.* s.l. : Journal of Object Oriented Programming, 1(3), 1988. pp. 26-49.

90. **Siamwalla, R., Sharma, R. and Keshav, S.** *"Discovering internet topology".* Cornell Univ. Ithaca, NY : s.n., 1999. Technical Report.

91. *"Topology Discovery in Heterogeneous IP Networks: The NetInventory System".* **Breitbart, Y., et al.** 3, June 2004, IEEE/ACM Transactions on Networking, Vol. 12, pp. 401-414.

92. *"Topology discovery for large Ethernet networks".* **Lowekamp, B., O'Hallaron, D.R. and Gross, T.R.** San Diego, CA, USA : ACM SIGCOMM, August 2001, pp. 237-248.

93. *IP Network Topology Discovery Using SNMP.* **Pandey, Suman, et al.** NJ, USA : s.n., 2009. 23rd international conference on Information Networking ICOIN'09. pp. 33-37.

94. *SNMP Object Navigator.* [Online] http://tools.cisco.com/Support/SNMP/do/BrowseOID.do?local=en&translate=Translate&objectInput=1.3.6.1.4.1.9.9.23.

95. Frame Formats - CDP Packet Formats. [Online] http://www.cisco.com/univercd/cc/td/doc/product/lan/trsrb/frames.htm#xtocid12.

96. Best Practices for Catalyst 4500/4000, 5500/5000, and 6500/6000 Series Switches Running CatOS Configuration and Management. [Online] http://www.cisco.com/en/US/products/hw/switches/ps4324/products_tech_note09186a0080094713.shtml#cdp.

97. LLDP MIB DEFINITIONS . [Online] http://www.ieee802.org/1/files/public/MIBs/LLDP-MIB-200505060000Z.txt.

98. SNMP Object Navigator - LLDP MIB. [Online] http://tools.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&step=2&mibName=LLDP-MIB.

99. *IEEE 802.1AB Station and Media Access Control Connectivity Discovery specified in standards document.*

100. W3C Recommendation XSL Transformations (XSLT) Version 1.0. [Online] November 16, 1999. http://www.w3.org/TR/xslt.

101. JUNG - Java Universal Network/Graph Framework. [Online] http://jung.sourceforge.net/.

102. **Bates, T., et al.** *IETF RFC 2858 Multiprotocol Extensions for BGP-4.* 2000.

103. **Rosen, E., Viswanathan, A. and Callon, R.** *IETF RFC 3031 Multiprotocol Label Switching Architecture.* 2001.

104. **Kent, S. and Seo, K.** *IETF RFC 4301 Security Architecture for the Internet Protocol.* 2005.

105. **Kent, S.** *IETF RFC 4303 IP Encapsulating Security Payload (ESP).* 2005.

106. **R. Stewart, Ed.** *IETF RFC 4960 Stream Control Transmission Protocol, http://www.ietf.org.* 2007.

107. **Monga, Pr. and Ahmed, I.** IPv6 integration and coexistence strategies for next generation networks. *Communications Magazine, IEEE.* Jan. 2004, Vol. 42, 1, pp. 88-97.

108. **Codd, Edgar F.** "A Relational Model of Data for Large Shared Data Banks.". *Communications of the Association for Computing Machinery.* pp. 377–87.

109. *"Constella: A Complete IP Network Topology Discovery Solution".* **Nazir, F., et al.** Sapporo, Japan : s.n., 2007. APNOMS 2007. pp. 425-436.

110. *"Analysis and visualization of network data using JUNG".* **Madadhain, J., et al.** 2, s.l. : Journal of Statistical Software, 2005, Vol. 10.

111. *"GraphML transformation", GD'04 Proceedings of the 12th international conference on Graph Drawing, Pages 89-99, Springer-Verlag Berlin, Heidelberg, 2004, ISBN:3-540-24528-6 978-3-540-24528-5.* **Brandes, Ulrik and Pich, Christian.** pp. 89-99.

112. *NAVY INTERNET PROTOCOL VERSION 6 (IPv6) TRANSITION STRATEGY IN SUPPORT OF NETWORK-CENTRIC OPERATIONS AND WARFARE.* **Nguyen, Ph. and Ferro, Robert.** s.l. : IEEE, 2008. MILCOM 2008. . pp. 1 - 7 .

113. **Case, J., et al.** *IETF RFC 1157 "A Simple Network Management Protocol (SNMP)".* May 1990.

114. **Hinden, R. and Deering, S.** *IETF RFC 3513 Internet Protocol Version 6 (IPv6) Addressing Architecture.*

115. **Mockapetris, P.** *IETF RFC 882 Domain Names - Concepts and Facilities.* November 1983.

116. —. *IETF RFC 883 Domain Names - Implementation and Specification.* November 1983.

117. **Huitema, C. and Carpenter, B.** *IETF RFC 3879 Deprecating Site Local Addresses.* September 2004.

118. **Malhotra, Ravi .** Enhanced Interior Gateway Routing Protocol (EIGRP). *IP Routing.* s.l. : O'Reilly & Associates, Inc., 2002, 4.

119. **Postel, J.** *IETF RFC 768 User Datagram Protocol.* August 28, 1980.

120. Generic Routing Encapsulation (GRE). [Online] http://www.cisco.com/en/US/tech/tk827/tk369/tk287/tsd_technology_support_sub-protocol_home.html.

121. **Dijkstra, E. W.** "A note on two problems in connexion with graphs". *Numerische Mathematik 1.* 1959, pp. 269–271.

122. **Coltun, R., et al.** IETF RFC 5340 OSPF IPv6. July 2008.

123. *IETF RFC 793 Transmission Control Protocol.* September 1981.

124. **Moy, J.** *IETF RFC 2328 OSPF Version 2.* April 1998.

125. **Oran, D.** *IETF RFC 1142 OSI IS-IS Intra-domain Routing Protocol.* February 1990.

126. **Bollapragada V., Wainner S., Khalid M.** *Advanced IPSec VPN Architecture and Design, Cisco Press.* 2005.

127. **Cabrera , Luis Felipe, Kurt, Christopher and Box, Don.** *An Introduction to the Web Services Architecture and Its Specifications.* [Online] http://msdn.microsoft.com/en-us/library/ms996441.aspx.

128. **Conta , A. and Deering, S.** *IETF RFC 2473 Generic Packet Tunneling in IPv6 Specification.* December 1998.

129. **Gilligan, R. and E. Nordmark.** *IETF RFC 2893 Transition Mechanisms for IPv6 Hosts and Routers.* August 2000.

130. **Guichard J., Pepelnjak I.** *Advanced IPSec VPN Architecture and Design, Cisco Press.* 2003.

131. Frequently Asked Questions About Java. [Online] http://www.oracle.com/technetwork/java/faq-141681.html#D.

132. Python - History and License. [Online] http://docs.python.org/license.html.

133. Perl Licensing. [Online] http://dev.perl.org/licenses/.

134. *IEEE 802.3 Ethernet.*

135. **Kent, S.** *IETF RFC 4302, IP Authentication Header.* 2005.

136. *IETF RFC 802.1D-2004 Media Access Control (MAC) Bridges.*

137. *ISO/ IEC 10589: 2002 Second Edition.*

138. *ITU-T Y.2000 Frameworks and functional architecture models.*

139. *ITU M.3190 Shared information and data model (SID).* 2008.

140. NoSQL. [Online] nosql-database.org.

141. OMG, Unified Modeling Language (UML).Version 1.4. [Online] 2002. http://www.omg.org/technology/documents/formal/uml.htm.

142. **Simpson, W.** *IETF RFC 1853 IP in IP Tunneling.* October 1995.

143. Web Services Description Language (WSDL) 1.1, W3C Note. [Online] March 15, 2001. http://www.w3.org/TR/wsdl.

144. XML Schema Part 0: Primer, W3C Recommendation. [Online] October 28, 2004. http://www.w3.org/TR/xmlschema-0/.

145. **Bass L., Clements P., Kazman R.** *Software Architecture in Practice.* s.l. : Addison Wesley, 2003.

146. **Garlan D., Monroe R., Wile D.** *ACME: An architecture description language.* s.l. : CASCON`97, 1997.

147. *Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation.* [Online] November 26, 2008. http://www.w3.org/TR/REC-xml/.

148. **Subharthi P., Raj J.,.** Architectures for the future networks and the next generation Internet: A survey. *Computer Communications.* 2011, Vol. 34, 1, pp. 2-42.

149. *A Novel DHT-Based Network Architecture for the Next Generation Internet.* **Hanka, O., Spleiss, C., Kunzmann, G., Eberspacher, J.** s.l. : ICN, 2009. pp. 332 - 341. ISBN: 978-0-7695-3552-4.

150. **Gilligan, R. and E. Nordmark,.** *IETF RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers.* 1996.

Списък на фигурите

Списък на таблиците