

Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 3, 812 19 Bratislava

Diagnostika porúch diskretných udalostných systémov založená na modeloch a aplikácie v informatike

Tímový projekt

Tím č. 5

Bc. Tomáš Bartek

Bc. Jozef Ferenčík

Bc. Lukáš Humený

Bc. Peter Kubanda

Bc. Pavol Kubík

Bc. Michal Kurtý

Bc. Michal Zapatický

Vedúci tímového projektu: Ing. Jana Flochová, PhD.

Zadanie

Diagnostika porúch diskretných udalostných systémov založená na modeloch a aplikácie v informatike.

V projekte ide o analýzu, programovanie a overovanie metód diagnostiky porúch diskretných udalostných systémov založenej na modeloch systému (konečný automat, stochastický automat, Petriho sieť atď.) a o navrhnutie a testovanie príkladov z oblasti informatiky, najmä počítačových sietí, architektúry softvéru, hardvéru a podobne.

Softvérový nástroj na analýzu a diagnostiku možnosti poruchy si študenti môžu jednak zvoliť (na Internete je mnoho voľne dostupného materiálu najmä na modelovanie diskretných systémov) alebo si ho môžu naprogramovať v C++, Visual C++, C#, Java, prípadne môžu prepísať niektoré modely i do VHDL.

Zoznam použitých skratiek

PN (Petri Nets) – Petriho siete

SPN (Stochastic Petri Nets) – Stochastické Petriho siete

PPN (Probability Petri Nets) – Pravdepodobnostné Petriho siete

P (Places) – Miesta

LAN (Local Area Network) – Lokálna sieť

T (Transitions) - Prechody

F (Fire) – Spustenie prechodu

MO (Initial marking) – Počiatočné značkovanie

CPN (Colored Petri net) – Farebné Petriho siete

1 Obsah

1	OBSAH	3
2	ÚVOD	7
2.1	PREHĽAD DOKUMENTU	7
3	ZÁKLADY MODELOVANIA A SIMULOVANIA SYSTÉMOV	8
3.1	KLASIFIKÁCIA SYSTÉMOV	8
3.1.1	<i>DETERMINISTICKÉ A NEDETERMINISTICKÉ (STOCHASTICKÉ) SYSTÉMY</i>	8
3.1.2	<i>SPOJITÉ A DISKRÉTNE SYSTÉMY</i>	9
3.1.3	<i>CENTRALIZOVANÉ A DISTRIBUOVANÉ SYSTÉMY</i>	9
3.2	MODEL SYSTÉMOV	10
4	PRINCÍPY MODELOVANIA A SIMULOVANIA SYSTÉMOV	11
4.1	KLASIFIKÁCIA MODELOV	12
4.2	SPÔSOBY POPISU SIEŤOVÝCH SYSTÉMOV	13
4.2.1	<i>KONEČNÝ AUTOMAT</i>	13
4.2.2	<i>DES</i>	13
4.2.3	<i>PETRIHO SIETE</i>	14
5	PETRIHO SIETE	15
5.1	TYPY PETRIHO SIETÍ	16
5.1.1	<i>C/E PETRIHO SIETE</i>	17
5.1.2	<i>P/T PETRIHO SIETE</i>	18
5.1.3	<i>DETERMINISTICKÉ ČASOVANÉ PETRIHO SIETE</i>	19
5.1.4	<i>STOCHASTICKÉ ČASOVANÉ PETRIHO SIETE</i>	19
5.1.5	<i>FAREBNÉ PETRIHO SIETE (CPN)</i>	20
5.2	VLASTNOSTI PETRIHO SIETI	22
5.2.1	<i>VLASTNOSTI SPRÁVANIA SA PS</i>	22
5.2.2	<i>VLASTNOSTI ŠTRUKTÚRY PS</i>	22
5.2.3	<i>DOSIAHNUTELNOSŤ ZNAČKOVANIA A POKRYTIE</i>	23
5.2.4	<i>OHRANIČENOSŤ A BEZPEČNOSŤ</i>	24
5.2.5	<i>ŽIVOSŤ A REVERZIBILNOSŤ</i>	24
5.2.6	<i>BEZKONFLIKTNOSŤ</i>	25
5.2.7	<i>KONZERVATÍVNOSŤ</i>	26
6	PROBLEMATIKA LOKALIZÁCIE PORÚCH V KOMUNIKAČNÝCH SYSTÉMOCH	27
6.1	DETEKCIA A IZOLÁCIA PORÚCH	27
6.2	TECHNIKY PRECHÁDZANIA MODELU	28
6.3	GRAFOVO TEORETICKÉ TECHNIKY	28
7	MODELOVANIE PETRIHO SIETÍ	29
7.1	PROGRAMOVÉ NÁSTROJE	30
7.1.1	<i>CPN TOOLS (COLOURED PETRI NETS)</i>	30

7.1.2	PLATFORM INDEPENDENT PETRI NET EDITOR 2 (PIPE2)	31
7.1.3	NETLAB	32
7.1.4	HPSIM	33
7.1.5	ZHODNOTENIE MODELOVACÍCH NÁSTROJOV	34
8	PROTOKOL OSPF	36
8.1	ZÁKLADNÁ FUNCIONALITA OSPF SMEROVAČOV	36
8.2	TYPY SIETÍ V OSPF	37
8.3	TYPY OSPF PAKETOV	37
8.4	TYPY SPRÁV LINK STATE ADVERTISEMENTS	39
8.5	VYTVÁRANIE SUSEDSTIEV MEDZI OSPF SMEROVAČMI	41
9	ŠPECIFIKÁCIA A NÁVRH RIEŠENIA	43
9.1	SIMULÁCIA VYBRANÝCH MECHANIZMOV PROTOKOLU OSPF	44
9.1.1	ČO CHCEME SIMULOVAŤ	44
9.1.2	AKO CHCEME SIMULOVAŤ	44
9.2	KONCEPCIA MODELU	45
9.3	PROTOTYP MODELU	47
9.3.1	PRECHOD ZO STAVU DOWN DO STAVU INIT	51
9.3.2	PRECHOD ZO STAVU INIT DO STAVU 2WAY	51
9.3.3	PRECHOD ZO STAVU 2 WAY DO STAVU EXSTART	52
9.3.4	PRECHOD ZO STAVU EXSTART DO STAVU EXCHANGE	52
9.3.5	PRECHOD ZO STAVU EXCHANGE DO STAVU LOADING/FULL	53
10	IMPLEMENTÁCIA MODELU V NÁSTROJI CPN TOOLS	54
10.1	KONCEPCIA MODELU SIETE S OSPF SMEROVAČMI	54
10.2	POPIS FUNKCIONALITY	55
10.2.1	PRIJÍMANIE A ODOSIELANIE PAKETOV	55
10.2.2	PRIDÁVANIE DO NBR DB	55
10.2.3	VOĽBA DR A BDR	55
10.3	HIERARCHICKÉ MODULY ARCHITEKTÚRY	56
10.3.1	MODUL PREPÁJACEJ SIETE	56
10.3.2	MODUL SMEROVAČA	59
10.4	POUŽITÉ ŠTRUKTÚRY A PREMENNÉ	64
10.5	POČIATOČNÉ ZNAČKOVANIE SIETE	66
10.6	KONCOVÉ ZNAČKOVANIE (SKONVERGOVANÁ SIEŤ)	66
11	OVEROVANIE FUNKČNOSTI V PROSTREDÍ CPN TOOLS	68
11.1	SIMULÁCIA MODELU	68
11.2	STATE-SPACE ANALÝZA	70
12	ZHODNOTENIE A BUDÚCA PRÁCA	76
13	POUŽITÁ LITERATÚRA	77
14	PRÍLOHY	3
14.1	DOKUMENTÁCIA RIADENIA	3
	HISTÓRIA DOKUMENTU	6
	ÚVOD	7

1 PONUKA	8
1.1 ZADANIE.....	8
1.2 MOTIVÁCIA	8
1.3 ANALÝZA ZADANIA A HRUBÝ NÁVRH RIEŠENIA	9
1.4 ČLENOVIA TÍMU	9
<i>BC. TOMÁŠ BARTEK</i>	10
<i>BC. JOZEF FERENČÍK</i>	10
<i>BC. LUKÁŠ HUMENÝ</i>	10
<i>BC. PETER KUBANDA</i>	11
<i>BC. PAVOL KUBÍK</i>	11
<i>BC. MICHAL KURTÝ</i>	11
<i>BC. MICHAL ZAPATICKÝ</i>	12
1.5 TÉMY PODĽA PRIORITY ZÁUJMU	12
1.6 STRETNUTIA TÍMU V ZS	12
1.7 STRETNUTIA TÍMU V LS.....	12
2.PLÁN PROJEKTU	13
2.1 PLÁN NA ZIMNÝ SEMESTER	13
2.2 PLÁN NA LETNÝ SEMESTER.....	14
3 ÚLOHY V TÍME	16
3.1 KOMUNIKÁCIA V TÍME.....	16
3.2 STRETNUTIE ČLENOV TÍMU MIMO PRAVIDELNÝCH STRETNUTÍ	16
3.3 MOBILNÝ TELEFÓN (SKYPE)	16
3.4 CHAT (ICQ)	16
3.5 GOOGLE GROUPS.....	16
3.6 E-MAIL.....	16
3.7 FUNKCIE ČLENOV TÍMU.....	17
3.8 DLHODOBÉ ÚLOHY	17
3.9 KRÁTKODOBÉ ÚLOHY	18
4 ZÁZNAMY O STRETNUTIACH	21
4.1 ZÁPISNICA 01	21
4.2 ZÁPISNICA 02	21
4.3 ZÁPISNICA 03	23
4.4 ZÁPISNICA 04	24
4.5 ZÁPISNICA 05	25
4.6 ZÁPISNICA 06	26
4.7 ZÁPISNICA 07	27
4.8 ZÁPISNICA 08	29
4.9 ZÁPISNICA 09	30
4.10 ZÁPISNICA 10	31
4.11 ZÁPISNICA 11	32
4.12 ZÁPISNICA 12	33
4.13 ZÁPISNICA 13	34

4.14 ZÁPISNICA 14	35
4.15 ZÁPISNICA 15	36
4.16 ZÁPISNICA 16	37
4.17 ZÁPISNICA 17	38
4.18 ZÁPISNICA 18	38
4.19 ZÁPISNICA 19	39
4.20 ZÁPISNICA 20	40

2 Úvod

Cieľom tohto tímového projektu je analyzovať a diagnostikovať možné poruchy v diskrétnych udalostných systémoch pomocou modelov Petriho sietí. Pre správne navrhnutie modelu nejakého systému je dobre o ňom mať čo najviac informácií. Je potrebné poznať jeho správanie v jednotlivých situáciách, potreby na riadenie, fyzikálne či iné obmedzenia daného procesu. Ďalším krokom je presne si zadefinovať aké sú požiadavky na daný systém, čo chceme aby systém vykonával. Dôležité je určiť si správne priority, aby nedochádzalo ku kolíziám. Podľa týchto požiadaviek sa pokúsime v tejto práci vytvoriť čo najlepší všeobecný model diskrétneho systému. Medzi najpoužívanejšie spôsoby vytvorenia modelov patria Konečné automaty a Petriho siete. Podrobnejšie sa v tejto práci budeme venovať Petriho sieťam, keďže pre tento spôsob modelovania sme sa rozhodli.

2.1 Prehľad dokumentu

V úvodných kapitolách analytickej časti dokumentu (kapitola 3 a 4) približujeme čitateľovi základné princípy a rozdelenia modelovania systémov, vysvetľujeme základné pojmy súvisiace s oblasťou modelovania a simulácie.

V ďalšej kapitole (kapitola 5) opisujeme problematiku Petriho sietí, ich základnú teóriu, rozdelenie a vlastnosti.

V kapitole č.6 stručne popisujeme princípy a techniky analýzy a lokalizácie porúch v simulovaných systémoch.

Kapitola č.7 približuje čitateľovi a porovnáva najrozšírenejšie programové prostriedky, pomocou ktorých prebieha simulácia a analýza navrhnutých modelov.

V kapitole č.8 je popísaná špecifikácia navrhovaného systému a funkcionality vybraného simulovaného sieťového protokolu - OSPF.

3 Základy modelovania a simulovania systémov

Modelovanie systému je činnosť, ktorá človeku umožňuje uvažovať o reálnom svete a na základe získaných poznatkov ho cieľavedome ovplyvňovať. Pod pojmom **systém** obvykle rozumieme abstrakciu reality, pričom sa zameriavame len na tie skutočnosti, ktoré sú relevantné pre naše skúmanie [1]. Keď hovoríme o modelovaní systému, znamená to, že ho reprezentujeme v nejakom inom prostredí (napríklad v jazyku alebo forme, ktorá je zrozumiteľná počítaču), kde je nevyhnutné uvažovať jeho vnútorné charakteristiky, ako je vzájomná interakcia medzi systémom a jeho okolím.

Z matematického hľadiska je to množina prvkov spojených navzájom s okolím za pomoci interakcií, čím sa vytvárajú vlastnosti umožňujúce plnenie účelových a cieľových funkcií. Štruktúrou sa rozumie vnútorné usporiadanie systému vyjadrené vzájomnými väzbami a pôsobením zložiek vnútornej organizácie celku, alebo jeho častí. Formálny popis systému je možné vyjadriť tvarom $S(X,R)$, kde X je množina prvkov a R je množina relácií alebo vzťahov medzi prvkami. Najjednoduchší systém je zostavený z dvojice prvkov, v tomto prípade množina R obsahuje iba jedinú reláciu – vzťah medzi dvoma prvkami. Všeobecne sa dá povedať, že každý prvok systému je sám systémom a označuje sa ako podsystém.

3.1 Klasifikácia systémov

V teórii systémov existuje viacero spôsobov, podľa ktorých je možné klasifikovať systémy, medzi najčastejšie používané patria delenia opísané v nasledujúcich kapitolách.

3.1.1 Deterministické a nedeterministické (stochastické) systémy

Hovoríme, že systém je **deterministický**, ak môžeme určiť jeho výstupné veličiny a stav v ľubovoľnom okamihu $t > t_0$ s istotou. To znamená, že jeho správanie je jednoznačne určené jeho stavom a podnetmi.

V **nedeterministickom** systéme sa výstupy nereprodukujú. Jeho výstupné veličiny a stav možno určiť len s určitou pravdepodobnosťou alebo inými štatistickými metódami. Medzi nedeterministické systémy patria tzv. systémy stochastické, teda systémy, v ktorých prebiehajú stochastické procesy ako väzby medzi jeho prvkami, pričom pojem **stochastický**

proces definujeme ako zmenu systému v čase podmienenou vplyvom nekontrolovateľných náhodných faktorov.

3.1.2 Spojité a diskretné systémy

Systém je označovaný ako **spojitý**, ak vstupné, výstupné a stavové veličiny sú definované pre každé t v určitom intervale $t_0 \leq t \leq t_k$ a správanie je opísané diferenciálnou, algebraickou, diferenčnou rovnicou alebo systémom rovníc, z čoho vyplýva, že dej prebieha vo všetkých prvkoch súčasne.

Systém je **diskretný**, ak vstupné, výstupné a stavové veličiny sú definované len v diskretných časových okamihoch t_n , obvykle $t = n.T$, kde n je postupnosť spravidla celých čísel z intervalu $t_0 \leq t_n \leq t_k$. U diskretných systémov pozorujeme len skokové zmeny stavu, spôsobené výskytmi udalostí [2]. Medzi vlastnosti diskretných systémov patrí najmä stabilita, možnosť riadenia a pozorovateľnosť, vďaka čomu sú najčastejšie používaným typom pri modelovaní väčšiny typov zložitých dynamických systémov, ako sú počítačové systémy, distribuovanej či paralelnej povahy, workflow systémy, sieťové systémy či komunikačné protokoly.

V súčasnosti sú v teórii systémov najrozšírenejšími objektmi skúmania a modelovania tzv. **systémy diskretných udalostí**. V tomto prípade je každý objekt reprezentovaný ako jednotka definovaná množinou premenných, kde ich hodnota v istom časovom úseku opisuje stav systému. Tak ako sa jednotlivé jednotky v čase menia, generujú tzv. udalosti, ktoré sú schopné modifikovať stav systému.

V teórii modelovania existujú premenné, ktoré môžu nadobudnúť nekonečnú množinu hodnôt a nazývajú sa **spojité premenné**. Ostatné premenné, tzv. **diskretné premenné**, môžu nadobudnúť hodnoty len z jednej konečnej množiny hodnôt. Príkladom spojitej premennej je čas príchodu a odchodu paketu v komunikačnej sieti, príkladom diskretnej premennej je počet uzlov, ktorými paket prejde. Podľa typu premenných použitých pre opis systému, hovoríme o **modeli so spojitými stavmi**, ak sú jeho premenné spojitej povahy, a v prípade, že použité premenné sú diskretné, hovoríme o **modeli s diskretnými stavmi**. Prepojenie týchto dvoch typov modelov predstavuje tzv. zmiešaný alebo hybridný model.

3.1.3 Centralizované a distribuované systémy

Distribuované systémy sú systémy, v ktorých sa jednotlivé časti procesov vykonávajú simultánne na dvoch alebo viacerých počítačoch, ktoré spolu komunikujú cez počítačové

sieť, zatiaľ čo **centralizovaný systém** spracúva procesy na jedinom počítači. Distribuované systémy sú špecifickým typom paralelných systémov.

3.2 Modely systémov

Pod pojmom **model** vo všeobecnosti rozumieme vytvorený systém, ktorý je určitým (niekedy i podstatným spôsobom) zjednodušením originálu modelovaného systému. Medzi originálom a jeho modelom existuje homomorfný vzťah zobrazenia, pričom rozlišujeme medzi modelmi **abstraktnými** (teda myšlienkovými, teoretickými), nad ktorými môžeme viesť logické úvahy a modelmi **simulačnými** (konkrétnymi, fyzicky realizovanými, spustiteľnými), na ktorých môžeme vykonávať simulačné experimenty.

Niektoré modely realizované prostredníctvom počítačov môžu patriť do oboch spomínaných kategórií súčasne, nakoľko poskytujú teóriu, umožňujúcu logickým odvodzovaním dokazovať vlastnosti modelu a zároveň je možné s týmto modelom vykonávať simulačné experimenty. Zatiaľ čo logické odvodzovanie vlastností umožňujú formálne modely založené na nejakom jednoduchom matematicky dobre spracovateľnom formalizme, počítačovú simuláciu umožňujú všetky vykonateľné modely, ktoré sú zapísané v nejakom programovacím jazyku.

Správanie dynamického systému môžeme charakterizovať stavovou premennou a zmenami jej hodnoty v čase, pričom mapovanie času na stav (priebeh hodnôt stavu v čase) nazývame procesom.

4 Princípy modelovania a simulovania systémov

Vytváranie modelov takých systémov, ktoré sú predmetom nášho skúmania, je veľmi rozšírenou činnosťou človeka už od počiatkov jeho existencie. Obecne sa dá povedať, že naša predstava sveta je vlastne modelom reality, teda okolitého sveta, ktorý môžeme skúmať. Pri modelovaní vychádzame z informácií o systéme, ktoré sú dostupné. Model, ktorý vznikne, reprezentuje formalizované znalosti o modelovanom systéme z hľadiska, ktoré chceme skúmať a obvykle pokrýva len tú časť popisu celého systému, ktorá je pre daný účel podstatná. Nakoľko model vždy vychádza z nejakej podmnožiny našich znalostí, ktoré sú neúplné, môžeme modelovať výlučne to, čo sme schopní pochopiť a opísať. Simulačné modelovanie teda predstavuje proces transformácie znalostí zo strojovo neakceptovateľnej reprezentácie na reprezentáciu akceptovateľnú počítačom.

Základnými etapami modelovania a simulácie sú [3]:

1. Vytvorenie abstraktného modelu – formovanie zjednodušeného popisu skúmaného systému
2. Vytvorenie simulačného modelu – zápis abstraktného modelu formou programu
3. Simulácia – experimentovanie s prezentáciou simulačného modelu
4. Analýza a interpretácia výsledkov - overenie správnosti modelu

Počítačový model je napodobenina systému iným systémom, teda počítačovým programom. Model systému musí napodobňovať všetky pre naše účely podstatné vlastnosti systému, nakoľko reprezentuje formalizované znalosti o modelovanom systéme z hľadiska, ktoré je predmetom skúmania.

Modelovanie je proces vytvárania modelu systému na základe určitých znalostí. Tento proces je obecne veľmi náročný a často vyžaduje znalosti z viacerých oborov. Kvalita vytvoreného modelu často zásadným spôsobom ovplyvňuje výsledky získané neskorším experimentovaním.

4.1 Klasifikácia modelov

Presné rozdelenie modelov do kategórií je pomerne náročné, pretože často neexistuje jednoznačné klasifikačné kritérium. Modely môžu byť delené napríklad podľa spôsobu ich matematického popisu, podľa úrovne abstrakcie, podľa metód implementácie (paralelné, distribuované) a podľa celej rady ďalších kritérií. V rámci potrieb tejto práce sa obmedzíme na klasifikačné kritérium podľa návrhu modelov:

- **Konceptuálne modely** – sú to neformálne popisy systémov, ktoré definujú základnú štruktúru systému (objekty a ich vzťahy), nedefinujú však kategórie bežné v teórii sieťových systémov (stav, udalosť, funkcia). Obvykle sa používajú v počiatočnej fáze modelovania a majú formu textov alebo obrázkov.
- **Deklaratívne modely** – modely, ktoré slúžia na popis prechodov medzi stavmi systému, pričom samotný model je definovaný stavmi a udalosťami, ktoré spôsobujú prechod z jedného stavu do druhého za istých podmienok. Vhodné sú predovšetkým pre diskretné modely a obvykle bývajú zapuzdrené do objektov, ktoré vytvárajú charakteristickú hierarchickú štruktúru. Patria sem konečné automaty (deterministické i nedeterministické - Markovove modely) či Petriho siete.
- **Funkcionálne modely** – modely zobrazené vo forme grafov, ktoré zobrazujú funkcie a premenné vo forme uzlov grafu. Medzi predstaviteľov tohto typu modelov patria systémy hromadnej obsluhy s frontami, kompartmentové systémy či grafy signálových tokov.
- **Modely popísané rovnicami** – modely, ktoré sú charakteristické svojim popisom rovnicami algebraickými, diferenciálnymi či diferenčnými alebo neorientovanými grafmi (elektrické schémy a tzv. Bond Graphs).
- **Priestorové modely** – tento typ modelov rozdeľuje systém na priestorovo menšie ohraničené podsystémy, ktoré majú definované správanie, cez ktoré dokážeme pochopiť komplexné fungovanie systému. Medzi ne patria tzv. celulárne automaty či L-systémy.
- **Multimodely** – medzi tento typ modelov patria modely, ktoré sú zložené z iných modelov a sú obvykle heterogénnej povahy, teda popísané rôznymi spôsobmi.

Patria sem kombinované modely (prepojenie spojitých a diskretných charakteristík), modely s neurčitostou (fuzzy modely) a prepojené simulačné systémy (HLA).

4.2 Spôsoby popisu sieťových systémov

Medzi základné spôsoby popisu sieťových systémov patria nasledovné formy:

4.2.1 Konečný automat

Konečný automat je zúžením všeobecného prechodového systému. Je to abstraktný model slúžiaci na modelovanie systémov, u ktorých je možné vymedziť konečný počet stavov a konečný počet vonkajších podnetov. Stav tohto systému sa zmení iba na základe vonkajšieho podnetu a to jednoznačne. Množiny vstupov, výstupov, stavov sú konečné. Od toho je odvodený pojem konečný automat. [2]

4.2.2 DES

DES (Diskretný udalostný systém) je charakterizovaný množinou stavov S , množinou asynchrónnych udalostí v čase E , ktoré zodpovedajú jednotlivým prechodom medzi stavmi a prenosovou funkciou:

$$D (D: S \times E \rightarrow S \cup \Lambda)$$

kde Λ indikuje nulový element, použitý na identifikáciu nedefinovaného prechodu).

Potom možno DES opísať nasledovne:

$$\Sigma = (S, E, D).$$

V prípade, že je známa informácia o výskyte k -tej udalosti e_k v čase t_k ($\{e_k, t_k\}_{k=0,1,\dots}$), hovoríme o tzv. **časových diskretných udalostných systémoch**. V takom prípade možno diskretný udalostný systém charakterizovať ako množinu:

$$\Sigma = (S, E, D, F)$$

kde

$$F = \{F_e(\cdot) : e \in E\}$$

je množina pravdepodobnosti distribučnej funkcie asociovanej typom udalosti. Náhodná premenná τ_e charakterizovaná $F_e(\cdot)$ je označovaná ako životnosť udalosti (*angl. lifetime*) a následne aj množina F je označená ako generátor životnosti udalostí daného DES. [4]

4.2.3 Petriho siete

Petriho sieť je matematická reprezentácia diskretných distribuovaných systémov. Graficky reprezentuje štruktúru distribuovaného systému ako orientovaný bipartitný graf s ohodnotením. Petriho siete sú nástrojom pre modelovanie a simulovanie diskretných javov. Je to grafová štruktúra, ktorá obsahuje dva druhy uzlov - miesta a prechody, ďalej obsahuje hrany a značky, ktoré sa vyskytujú v miestach. Simulácia v Petriho sieti prebieha pomocou premiestňovania tokenov z miesta do miesta, ak to dovoľuje prechod. Bližšie informácie ohľadom Petriho sietí rozoberáme v kapitole (5).

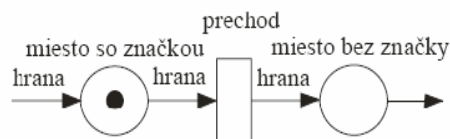
5 Petriho siete

Petriho siete (Petri Nets, ďalej PN) sú vhodným grafickým, ale aj matematickým nástrojom na modelovanie, analýzu a verifikáciu vlastností diskretných systémov, ktoré môžu byť [1]:

- Súbežné (concurrent)
- Asynchrónne
- Paralelné
- Distribuované
- Stochastické (nedeterministické)

PN predstavuje orientovaný bipartitný graf obsahujúci dva typy uzlov: **miesta** (places) a **prechody** (transitions), ktoré sú vzájomné poprepájané **hranami** (arcs).

Prívlastok **orientovaný** znamená to, že hrany v grafe sú orientované. Prívlastok **bipartitný** označuje stav, kde sa množina uzlov grafu skladá z dvoch vzájomne disjunktných podmnožín (množina miest a množina prechodov). Pri grafickej reprezentácii PN sa miesta znázorňujú ako krúžky a prechody ako obdĺžniky. Hrany spájajú miesto s prechodom alebo prechod s miestom. **Značenie** (marking), resp. stav priradí každému miestu nezáporné celé číslo. Ak značenie priradí miestu p nezáporné celé číslo k , hovoríme, že p je označené (marked) k značkami - **tokenmi** (tokens). Pri grafickej reprezentácii PN umiestnime do krúžku symbolizujúceho dané miesto k bodiek. Jednotlivé prvky PN sú zobrazené na obrázku č.1, pričom platí, že:



Obrázok č.1: Komponenty zovšeobecnenej Petriho siete.

- miesto môže obsahovať celočíselný nezáporný počet tokenov
- v okamihu aktivácie prechodu sa odoberú tokeny zo vstupných miest a pridajú sa na výstupné miesto prechodu
- orientované hrany prepájajú miesta a prechody
- počiatočné značenie (umiestnenie tokenov v miestach pred prvým preskokom) popisuje počiatočný stav systému

- vývoj systému je reprezentovaný presunom tokenov v sieti na základe aktivácie (prepálenia) prechodu
- každé nové značenie predstavuje nový stav [1].

5.1 Typy Petriho sietí

Existuje viacero rôznych rozšírení (modifikácií) Petriho sietí, ktoré boli vytvorené aby boli schopné jednoduchšie modelovať viacero oblastí praktického využitia. Petriho siete boli a stále sú upravované tak, aby vyhovovali potrebám systémov na ktoré nestačí opis iba pomocou základných Petriho sietí. Tieto zmeny a prídavné rozšírenia Petriho sietí nám slúžia na jednoduchšie a zrozumiteľnejšie modelovanie a opis zložitých a komplexných dynamických systémov.

Typy Petriho sietí rozlišujeme na dve základné skupiny a to PS nízkej úrovne a PS vyššej úrovne.

Petriho siete nízkej úrovne:

- C/E (Condition/Event) Petriho siete,
- P/T (Place/Transitions) Petriho siete,
- Deterministické časované (Timed) Petriho siete,
- Stochastické časované (Stochastic) Petriho siete.

Petriho siete vyššej úrovne:

- Farebné (Coloured) Petriho siete,
- Hierarchické (Hierarchical) Petriho siete,
- Fuzzy Petriho siete,
- Objektovo-orientované (Object-Oriented) Petriho siete.

Rôzne rozšírenia Petriho sietí môžu byť vzájomne kombinované aby sme dosiahli čo najlepšiu modelovaciu schopnosť. Petriho siete vyššej úrovne môžu byť napríklad rozšírené o časovú závislosť. Fuzzy Petriho siete sú založené na teórii fuzzy množín. Časovanú fuzzy Petriho sieť chápeme ako model neurčitých časových súvislostí. Tento typ siete tu nebude bližšie rozoberaný [5].

5.1.1 C/E Petriho siete

C/E sieť je najjednoduchší typ, ktorý sa dokonca v niektorých prípadoch chová ako konečný stavový automat. Je podtriedou P/T Petriho sietí. C/E - Condition/Event (podmienka/udalosť) znamená, že sú tieto siete modelované podmienkami a udalosťami, ktoré môžu nastať pri ich splnení.

Základné pravidlá:

- miesta siete reprezentujú určité podmienky,
- prechody siete reprezentujú udalosti, ktoré môžu nastať iba ak sú splnené všetky vstupné podmienky a zároveň nie sú splnené žiadne výstupné podmienky,
- po prevedení udalosti platí, že sú splnené všetky jej výstupné podmienky a nie je splnená žiadna vstupná podmienka,
- značky siete sa nachádzajú v miestach a reprezentujú platnosť podmienky (logický stav), pričom v každom mieste môže byť len jedna alebo žiadna značka,
- orientované hrany reprezentujú tok udalostí.

Celkový stav siete je daný množinou podmienok splnených v danom časovom okamihu. K zmenám stavov siete dochádza uskutočňovaním udalostí. Ak má udalosť splnené vstupné podmienky ešte to neznamena, že v tom okamihu musí nastať, tj. môže ale nemusí. Pri udalostiach rozlišujeme či sa vyskytujú nezávisle na sebe alebo v určitom poradí.

Špeciálne prípady C/E Petriho sietí:

1. *Konečný automat* - každý prechod siete má práve jednu vstupnú a jednu výstupnú podmienku a v sieti sa nachádza jediná značka, ktorá určuje aktuálny stav automatu,
2. *Čistá sieť* - sieť v ktorej žiaden prechod neobsahuje slučku (loop), čo je miesto, ktorého vstupná aj výstupná hrana patrí tomuto prechodu (spustením udalosti sa nemení platnosť podmienky),
3. *Elementárna sieť* - je to čistá sieť v ktorej má každý prechod aspoň jednu vstupnú a jednu výstupnú podmienku [5].

5.1.2 P/T Petriho siete

P/T - Place/Transition (miesto/prechod), znamená že sú definované miestami (stavy) a prechodmi (udalosti).

Pravidlá P/T sietí:

- miesta siete môžu mať udanú kapacitu, ktorá úvádza maximálny možný počet značiek nachádzajúcich sa v danom mieste a je daná celým prirodzeným nezáporným číslom,
- ak miesto nemá udanú kapacitu má automaticky neobmedzenú kapacitu počtu značiek, hodnotu značenia miesta siete vyjadrujeme grafickým znázornením bodiek alebo pri väčšom počte celým nezáporným číslom,
- hrany siete môžu mať udanú váhu, ktorá udáva násobnosť hrany a je rovnako daná celým nezáporným prirodzeným číslom,
- ak hrana nemá udanú váhu má automaticky váhu rovnú 1, prechod je spustiteľný ak každé jeho vstupné miesto obsahuje minimálne taký počet značiek aká je váha jeho hrany vstupujúca do daného prechodu a každé jeho výstupné miesto má taký počet značiek, že jeho zvýšenie o váhu jeho hrany vystupujúcu z prechodu neprevyšuje danú kapacitu miesta,
- po spustení prechodu sa z každého vstupného miesta presunie toľko značiek aké sú váhy ich výstupných hrán a naopak do výstupných miest prechodu sa pridajú značky zodpovedajúce váham hrán vstupujúcim do týchto miest.

P/T Petriho siete s prioritami:

- Každému prechodu siete môže byť priradená priorita celým nezáporným číslom. Spustenie prechodu má tak ďalšiu vstupnú podmienku. Súčasne môžu byť spustené len prechody s rovnakou prioritou inak je vždy spustený prechod s vyššou prioritou.

P/T Petriho siete s inhibítormi:

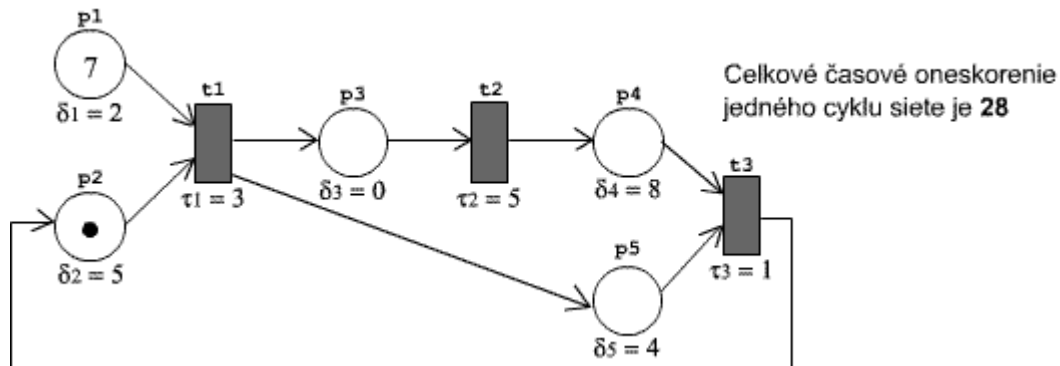
- Inhibítor je ďalší typ orientovanej hrany, ktorý je zakončený namiesto šípky kolieskom. Ich použitím podstatne zväčšujeme modelovaciú schopnosť Petriho siete až na úroveň Turingových strojov ale zároveň znižujeme možnosť ich analýzy. Je to vstupná testovacia hrana (podmienka) spustiteľnosti prechodu, ktorá je splnená ak miesto obsahuje menší počet značiek ako je váha inhibítora. Pri spustení prechodu sa z miesta, ktoré je s ním spojené inhibítorom neodoberajú žiadne značky narozdiel od štandardného vstupného miesta. Inhibítor je vždy možné použiť výhradne ako vstupnú hranu prechodu a nie ako výstupnú. P/T siete s inhibítormi sú z teoretického hľadiska schopné modelovať všetko čo je možné vyjadriť algoritmom [5].

5.1.3 Deterministické časované Petriho siete

Zapojením času do P/T Petriho sietí získame veľmi silný nástroj pre modelovanie dynamických systémov a ich analýzu výkonnosti. Čas je v prípade deterministických Petriho sietí vždy konštantný. Čas môže byť priradený všetkým prvkom Petriho siete a platí:

- priradenie času prechodom - po spustení prechodu zostávajú značky v prechode daný časový interval (pokiaľ daná udalosť neskončí),
- priradenie času miestam - značky zostávajú daný časový interval vo vstupnom mieste prechodu od jeho spustiteľnosti po dobu trvania prechodu,
- priradenie času značkám - značky majú po prevedení prechodu priradenú určitú časovú hodnotu počas ktorej nemôžu byť znova použité,
- priradenie času hranám - hranám je priradený daný časový interval prenosu značiek (vyjadruje rýchlosť prenosu) [3].

Na obrázku č.2 je sieť, v ktorej sú značkám a prechodom priradené časové oneskorenia. Časy pri miestach siete v tomto prípade reprezentujú časovú hodnotu počas ktorej nemôžu byť značky znova použité. Čas sa začína odraťovať hneď po vstupe jednotlivých značiek do daného miesta. Sieť vykonáva opakovane 7-krát za sebou rovnaký cyklus udalostí pričom celkové oneskorenie jedného cyklu je 28.



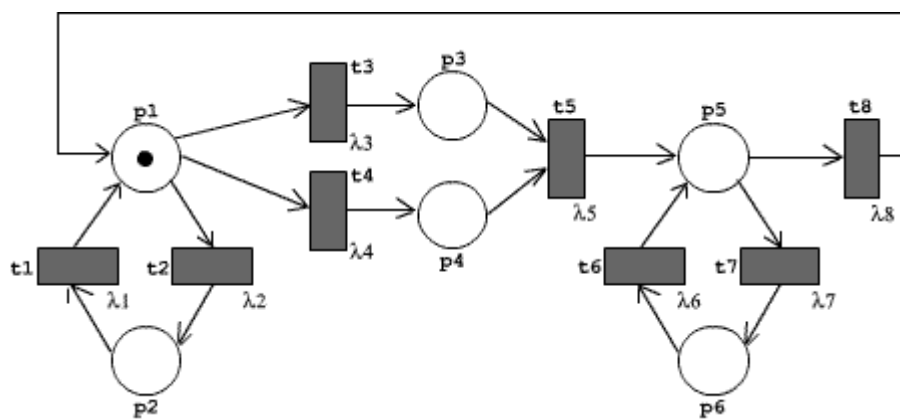
Obrázok č.2: Deterministická časovaná Petriho sieť.

5.1.4 Stochastické časované Petriho siete

Stochastický proces je taký proces v ktorom sú zmeny stavov podmienené vplyvom náhodných faktorov. V prípade stochastických Petriho sietí sú prechodom priradené početnosti a oneskorenia spúšťania, ktoré reprezentujú určité náhodné veličiny. Časové oneskorenie môže byť priradené aj miestam siete. Početnosti spúšťania prechodov majú náhodné exponenciálne

rozloženie a vyjadrujú ako často je spustený prechod ak má stále splnené vstupné podmienky. Prevrátené hodnoty početností prechodov vyjadrujú čas vykonania udalostí, ktoré tieto prechody reprezentujú. Počas vykonávania udalosti (prechodu), ktorý trvá náhodný čas čakajú značky vo vnútri prechodu. Stochastické PS sú rovnako využívané na analýzu výkonnosti systémov [5].

Na obrázku č.3 je stochastická Petriho sieť s priemernými početnosťami priradenými všetkým prechodom. Trvanie určitého prechodu je náhodná premenná s priemernou hodnotou rovnou $1 / \lambda_i$. Každý prechod predstavuje činnosť, ktorá trvá náhodný čas, počas ktorého sú značky zo vstupných miest blokované v prechode.



Obrázok č.3: Stochastická časovaná Petriho sieť.

5.1.5 Farebné Petriho siete (CPN)

Ak chceme opísať reálny systém, ktorý pracuje so zložitými a viacerými dátovými typmi, potom nám už základné Petriho siete ako model nestačia lebo by boli príliš zložité a ťažkopádne a takýto opis je veľmi náročný. Pre opis zložitých manipulácií s dátami boli vyvinuté vysokoúrovňové Petriho siete z ktorých sú najznámejšie a najpoužívanejšie farebné Petriho siete. Hlavnou myšlienkou je vzájomný pohyb viacerých typov farebne rozlíšených značiek v sieti.

Rozšírenia oproti P/T Petriho sieťam:

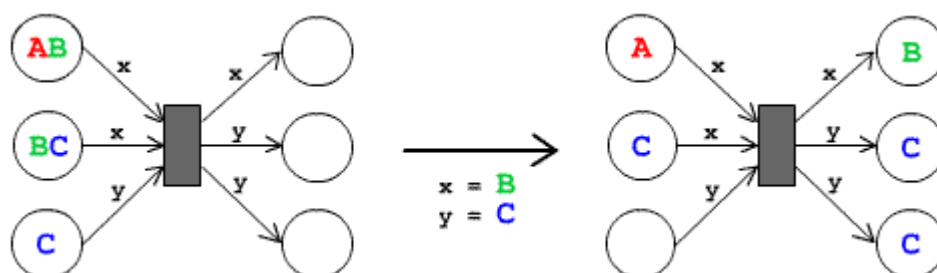
- sú rozšírené o viacero rôznych typov značiek siete, ktoré sú dané triedami farieb - sú farebne rozlíšené a značky rovnakej farby sú značky rovnakého typu,
- triedy farieb chápeme ako dátové typy, ktoré môžu byť konečné, nekonečné, jednoduché (elementárne) alebo zložené, (napr. výpočtové typy, rôzne druhy čísel, deň, rok, vektory,.. atď.), pričom každý dátový typ je daný množinou hodnôt,
- s dátovými typmi sú spojené najrôznejšie operácie, napr. číselné alebo logické operácie,

- každému miestu sú priradené typy značiek (triedy farieb), ktoré sa môžu v danom mieste nachádzať a stav miesta je daný multimnožinou jeho značiek, ktoré sa v ňom práve nachádzajú,
- každý prechod obsahuje navyše podmienku reprezentovanú výrazom vytvoreným z konštánt alebo premenných, ktorý po vyhodnotení dáva logickú hodnotu,
- z konštánt a premenných je taktiež vytvorený hranový výraz priradený hranám, ktorý po vyhodnotení dáva multimnožinu značiek typov prislúchajúcu miestu, ktoré patrí hrane.

Pre prechod farebnej PS platí:

- je spustiteľný ak je splnená jeho podmienka, ktorá po vyhodnotení dáva logickú hodnotu true,
- je spustiteľný ak multimnožina značiek každého vstupného miesta je rovnaká alebo väčšia ako multimnožina, ktorú dostaneme vyhodnotením hranového výrazu danej hrany, ktorá vedie z nášho vstupného miesta do prechodu,
- po jeho prevedení sa multimnožiny značiek vstupných miest zmenšia o multimnožinu, ktorú dostaneme vyhodnotením hranového výrazu vstupných hrán prechodu a naopak multimnožiny značiek výstupných miest sa zväčšia o multimnožinu, ktorá vznikne vyhodnotením hranového výrazu výstupnej hrany prechodu do daného miesta,
- pri oboch vyhodnoteniach hranového výrazu musia byť za premenné všade dosadené rovnaké hodnoty,
- prechod nemusí byť spustiteľný pri všetkých hodnotách dosadzovaných za premenné.

Jednoduchý príklad modelovania farebnej PS v ktorej sú hranám priradené premenné (x , y) a v miestach siete sú tri typy značiek alebo triedy farieb (A, B, C). K jednotlivým premenným priradíme typ značky podľa obrázka $x=B$, $y=C$. Ak by sme premenným priradili inú hodnotu napr. $x=A$, $y=C$ prechod by nebol spustiteľný lebo by neboli splnené vstupné podmienky. Po spustení prechodu sa značky presunú do výstupných miest a výsledné značkovanie vidíme na obrázku č.4 [5].



Obrázok č.4: Jednoduchá farebná Petriho sieť.

5.2 Vlastnosti Petriho siete

Reprezentujú dôležité problémy a črty základných P/T (Place-Transition) sietí riešené u všetkých formálnych modelov. Pomocou P/T Petriho sietí môžeme modelovať pomerne širokú triedu problémov, a zároveň môžeme takéto modely jednoducho analyzovať a určovať ich vlastnosti. Aby sme vedeli model systému vhodne analyzovať, musia byť jeho vlastnosti rozhodnuteľné. Teoretické podklady sú čerpané z [5].

5.2.1 Vlastnosti správania sa PS

Opisujú Petriho sieť z hľadiska jej správania a závisia od počiatočného značkovania siete M_0 .

- dosiahnuteľnosť značkovania
- ohraničenosť
- bezpečnosť
- živosť
- pokrytie
- bezkonfliktnosť
- bezkontaktnosť

Mnohé dôležité vlastnosti Petriho sietí možno analyzovať s využitím grafov dosiahnuteľnosti a pokrytia. Graf dosiahnuteľnosti Petriho siete je späť s dosiahnuteľnosťou.

5.2.2 Vlastnosti štruktúry PS

Opisujú Petriho sieť z hľadiska jej štruktúry a nezávisia od počiatočného značkovania siete M_0 . Štruktúrou je myslený návrh (konštrukcia) siete - poprepájanie základných stavebných prvkov PS a počiatočné označkovanie.

- reverzibilitnosť (konzistentnosť)
- konzervatívnosť

Ďalšie možnosti štruktúrnej analýzy Petriho sietí predstavujú metódy, ktoré sú založené na lineárnej algebraickej reprezentácii, použití matíc a riešení sústav rovníc. Sú to P-invarianty a T-invarianty a ide o štruktúrne vlastnosti. P-invariant je množina miest, ktoré nemenia svoje značky v priebehu spúšťania prechodov. Ich identifikovanie nám pomáha v analýze živosti PS a v

získovaní určitých špecifických vlastností, ktoré má modelovaný systém zachovávať. T-invariant udáva pri určitom značkovaní PS koľkokrát je potrebné spustiť každý prechod siete aby sme znovu dosiahli toto značenie. Maticová reprezentácia zjednodušuje prácu s pojmami, ktoré súvisia so značkovaním PS, a ktoré sa úzko dotýkajú chovania modelovaného systému. Tieto metódy si tu len veľmi stručne opíšeme.

5.2.3 Dosiahnuteľnosť značkovania a pokrytie

Počiatkové značkovanie siete M_0 je dané definíciou Petriho siete. Každý diskretný udalostný systém začína svoju činnosť určitým počiatkovým stavom. Otázka či je značenie dosiahnuteľné z počiatkového značenia, teda či je prvkom stavového priestoru PS, patrí medzi základné problémy analýzy Petriho sietí.

Značkovanie M_i je dosiahnuteľné zo značkovania M_j , keď M_i získame M_j realizáciou konečného počtu k priechodov. Znamená to, že existuje určitá postupnosť spustenia prechodov pri ktorej sa dostaneme z jedného značkovania siete do iného. V prípade že, $k = 1$, hovoríme o bezprostrednej dosiahnuteľnosti. Množinu všetkých značkování dosiahnuteľných z M_0 označujeme ako $R(M_0)$.

Graf dosiahnuteľnosti

Množinu dosiahnuteľných značkování a postupnosti spúšťania prechodov pre danú Petriho sieť vieme vizualizovať graficky pomocou orientovaného grafu. Slúži ako prehľadná pomôcka pri analýze viacerých vlastností Petriho siete. Graf má nasledujúce vlastnosti:

- množina $R(M_0)$ je množina všetkých vrcholov grafu,
- hrany reprezentujú spustenie konkrétneho prechodu,
- pri dosiahnutí duplicitného značkovania vedie hrana grafu k prvému výskytu tohto značkovania (systém sa tzv. vráti do určitého značkovania v ktorom sa už nachádzal),
- M_0 je počiatkový vrchol grafu.

Strom dosiahnuteľnosti

Postup vytvárania je podobný, s tým rozdielom, že pri dosiahnutí duplicitného značkovania sa tento vrchol označí ako koncový inou farbou a nepokračuje z neho žiadna hrana. Na začiatku máme množinu obsahujúcu jeden prvok a to značkovanie M_0 . Následne do množiny pridáme všetky bezprostredne dosiahnuteľné značkovania k značkovaniam, ktoré už máme v množine. Tretí krok je zistenie, či sa nám množina pridaním určitého značkovania zväčšila. Ak áno tak postupujeme znovu druhým krokom a ak nie znamená to, že sa značkovanie opakuje a nepokračujeme ďalej.

Pokrytie

Značkovanie M_i je pokryté značkováním M_j , ak je od neho menšie alebo mu je rovné, tj. platí $M_i \leq M_j$.

Určenie množiny dosiahnuteľných značkovaní pre neohraničenú (nekonečnú) Petriho sieť konečným spôsobom je tiež možné. Množina dosiahnuteľných značkovaní môže byť nekonečná, čo nastáva ak niektoré miesta v sieti môžu neobmedzene zväčšovať svoj počet zančiek (tokenov). Na vyjadrenie nekonečnosti sa používa symbol ω .

Graf pokrytia (strom pokrytia)

Pre neohraničené PS môžeme skonštruovať graf alebo strom pokrytia namiesto grafu dosiahnuteľnosti. Umožňujú reprezentovať množinu všetkých dosiahnuteľných značkovaní pomocou konečného počtu rozšírených značkovaní, ktoré pokrývajú všetky dosiahnuteľné značkovania v danej PS.

5.2.4 Ohraničenosť a bezpečnosť

Ohraničenosť miesta a siete

Pri opise správania sa systémov zisťujeme, či má špecifikovaný systém konečnú množinu všetkých stavov. V Petriho sieťach hovoríme o tejto vlastnosti ako o ohraničenosti siete. Miesto p je ohraničené vtedy, ak jeho počet značiek neprekročí určitú konečnú hodnotu k (k -ohraničené miesto). PS je k -ohraničená ak pri každom dosiahnuteľnom značkovaní sú všetky jej miesta maximálne k -ohraničené. To tiež znamená, že PS je ohraničená len ak je jej množina dosiahnuteľných značkovaní konečná. Ohraničenosť siete je požadovaná pri väčšine modelovaných systémov.

Bezpečnosť miesta a siete

PS sa nazýva bezpečná, ak je 1-ohraničená. To znamená, že počet značiek vo všetkých miestach siete musí byť 0 alebo 1 pri všetkých dosiahnuteľných značkovaniach. Bezpečnosť siete je požadovaná zväčša pri modelovaní logických podmienok v systémoch.

5.2.5 Živosť a reverzibilitnosť

Jedným zo základných problémov pri modelovaní udalostných systémov je vyvarovať sa takému stavu systému z ktorého nie je možné spustiť žiaden prechod PS čo znamená že je systém nečinný a nemôže v ňom nastať žiadna udalosť. Tento stav nazývame mŕtvý stav alebo uzamknutie (deadlock). Nastáva napríklad v situácii pri zdieľaní zdrojov, kde proces čaká na pridelenie zdroja, ktorý už bol pridelený inému procesu a podobne tento proces čaká na

pridelenie zdroja, ktorý už bol pridelený prvému procesu, čo znamená, že sa navzájom blokujú a systém sa uzamkne.

Živosť

PS je živá ak je každý jej prechod živý a pri činnosti systému, tj. pri spúšťaní prechodov siete, je vždy možné každý prechod spustiť znovu - nestráca možnosť, že by už nebol v budúcnosti spustený. Sieť je živá ak neobsahuje žiaden mŕtvy stav. Živosť siete závisí od počiatočného značkovania M_0 . Dosiadnutelné značkovanie je mŕtve ak z neho nie je možné spustiť žiaden prechod. Ak sú všetky dosiadnutelné značenia z M_0 živé, tak aj PS je živá.

Prechod v Petriho sieti je živý vtedy, ak pre každé jeho značkovanie M_i , dosiadnutelné z M_0 existuje také značkovanie M_j dosiadnutelné z M_i , z ktorého je priechod t spustiteľný.

Rozlišujeme viacero úrovní živosti prechodu PS:

- **L0-živý** (mŕtvy) : ak nie je spustiteľný v žiadnom z dosiadnutelných značkování,
- **L1-živý** : pokiaľ môže byť spustený aspoň raz - aspoň v jednom zo značkování,
- **L2-živý** : ak pre určité celé kladné číslo n môže byť v nejakej postupnosti značkování spustení aspoň n -krát,
- **L3-živý** : ak je spustiteľný nekonečný počet krát v určitej postupnosti značkování,
- **L4-živý** (živý) : ak pre každé zo značkování existuje postupnosť spustenia prechodov, pričom daný prechod je v postupnosti spustený aspoň raz.

PS je L_i -živá, ak sú všetky jej prechody L_i -živé.

Reverzibilnosť

Táto vlastnosť vyjadruje cyklický charakter správania sa systému. Veľa systémov totiž pracuje opakovaným vykonávaním určitej postupnosti udalostí (procesov). To znamená, že po ukončení vykonávanej postupnosti vždy vráti systém do počiatočného stavu pre opakovanie.

PS je reverzibilná ak je vždy dosiadnutelné počiatočné značkovanie. Platí, že z každého dosiadnutelného značkovania siete je dosiadnutelné počiatočné značkovanie M_0 .

Cyklická vlastnosť správania sa je využívaná najmä vo výrobných systémoch.

5.2.6 Bezkonfliktnosť

Konflikt v PS nastáva, ak spustenie jedného prechodu siete znižuje stupeň spustiteľnosti iného prechodu siete alebo jeho spustenie tohto prechodu úplne znemožňuje. PS nazývame

bezkonfliktnou ak v nej nemôže nastať žiadny takýto konflikt, tj. nemôže sa nachádzať v stave v ktorom sú viaceré prechody spustiteľné ale spustiteľnosť jedného zníži spustiteľnosť iného.

Rozlišujeme dva typy konfliktu:

- **symetrický** - ak prevedením jedného prechodu znížim spustiteľnosť druhého a aj naopak prevedením druhého ovplyvním prvý,
- **asymetrický** - ak prevedenie jedného prechodu znižuje spustiteľnosť druhého ale druhý neovplyvňuje spustiteľnosť prvého.

5.2.7 Konzervatívnosť

Časté použitie PS je aj pri modelovaní pridelenia rôznych systémových zdrojov v systéme. PS je striktno konzervatívna, ak celkový počet zdrojov (značiek) v počiatočnom značkovaní je rovnaký aj vo všetkých ostatných dosiahnuteľných značkovaniach. Počet značiek v PS ostáva po celú dobu činnosti systému nemenný.

PS je konzervatívna vzhľadom na váhový vektor v , ak platí, že súčin $M_0.v = M_1.v = \dots = M_j.v$, kde (M_1, \dots, M_j) je množina všetkých dosiahnuteľných značkovaní danej siete. O takejto PS môžeme hovoriť napríklad ak z jedného stavu (procesu) siete vzniknú po spustení prechodu dva a viac stavov (procesov), ktoré sa znovu spoja do jedného spoločného procesu.

Táto vlastnosť býva modelovaná najmä v distribuovaných počítačových systémoch, kde vzniká problém dostupnosti procesora a iných častí počítača a problém vzniku kritických sekcií.

6 Problematika lokalizácie porúch v komunikačných systémoch

Diagnostika, detekcia porúch a ich separácia sú veľmi dôležitými a náročnými úlohami v oblasti automatického riadenia veľkých komplexných systémov. Tieto úlohy v minulosti boli a v súčasnosti stále sú výzvou pre mnohých výskumných pracovníkov v oblasti teórie systémov a ich riadenia, prípadne tiež návrhu.

Komplexnosť komunikačných sietí spôsobuje, že prevencia porúch v zmysle zamedzenia ich vzniku a výskytu nie je možná (porúch hardvérového i softvérového charakteru, ale rovnako ako aj porúch spôsobených ľudskými chybami) – o to je dôležitejšia práve rýchla detekcia vzniknutých porúch, ich presná lokalizácia a následná izolácia/odstránenie. Rozsiahlosť existujúcich komunikačných sietí navyše vnáša potrebu maximálnej automatizácie tohto procesu.

Lokalizácia porúch funguje na základe zberania, klasifikácie a analýzy chýb. Vhodne fungujúci detekčný systém dokáže z rôznej konštelácie zaznamenaných chýb identifikovať a lokalizovať poruchu, ktorá je inak v mieste výskytu nebadateľná, respektíve nie je možné ju priamo lokalizovať na základe napríklad samostatných stavových informácií. [6]

Na riešenie takýchto úloh boli navrhnuté viaceré rôzne modely, prístupy a formalizmy, napr. stavovo priestorové metódy (state-space), metódy detekcie a izolácie porúch, poruchové stromy (fault trees), invariantov, použitie prostriedkov umelej inteligencie, a i.

6.1 Detekcia a izolácia porúch

Najpoužívanejšie techniky sú z tejto kategórie – sú najzrozumiteľnejšie a najjednoduchšie na pochopenie. Ideou je imitácia postupu skutočného človeka, ktorého schopnosti vychádzajú na jednej strane z hlbokých vedomostí o komunikačných sieťach a na druhej z praktických skúseností nadobudnutých praxou. Lokalizačné systémy najčastejšie používajú informačnú základňu reprezentovanú množinou pravidiel, ktoré sa v cykloch zväzujú do postupností podľa aktuálnych posudzovaných podmienok. V systémoch s malým rozsahom môže byť táto technika účinná – skúsenejší profesionál dokáže vymyslieť množinu pravidiel, ktorá pokryje najčastejšie poruchy (čiže ide viac o „praktický“ prístup, ako „vedecký“).

6.2 Techniky prechádzania modelu

Tieto techniky využívajú formálnu reprezentáciu komunikačného systému s jasne definovanými vzťahmi a spojeniami medzi komponentami siete (entitami tvoriacimi danú komunikačnú sieť). Proces lokalizácie poruchy začína v mieste hlásenej chyby (zistenej nezrovnalosti zariadením v komunikačnej sieti v monitorovaných veličinách a stavoch), nasleduje identifikovanie všetkých ďalších súvisiacich chýb a pomocou takto vytvoreného súboru sa identifikujú poruchové elementy v sieti.

6.3 Grafovo teoretické techniky

Tieto techniky sú postavené na grafovom modeli systému, konkrétne na smerovom modeli propagácie porúch, ktorý popisuje, ako a kde sa konkrétna porucha prejaví. Zostavenie takéhoto modelu vyžaduje presné a podrobné informácie o závislostiach medzi jednotlivými komponentami siete a od jeho presnosti sú následne priamo úmerne závislé aj výsledky lokalizácie porúch. V praxi sa rozsiahlosť redukovať potrebou modelov propagácie porúch len pre práve existujúce spojenia. Zaujímavá je možnosť viacerých typov porúch jedného objektu v prípade použitia kauzálnych grafov namiesto závislostných (napríklad kompletne znefunkčnenie, dlhá odozva, vysoká stratovosť paketov...) - takto popisované stavy sa približujú reálnemu systému fungujúcemu v reálnom svete oveľa viac, ako predchádzajúce techniky, žiaľ opäť za cenu citeľne zložitejšieho návrhu a zostavenia modelu.

7 Modelovanie Petriho sietí

Cieľom nášho tímového projektu je vytvoriť model sieťového protokolu. Vytváranie komplexného modelu Petriho siete bez pomoci softvérových aplikácií v dnešnej dobe je ťažko realizovateľné. Potreba vytvoriť presný, stabilný a dostatočne funkčný model nás priviedla k výberu vhodného programového nástroja.

Hlavnými kritériami sú pre naše účely práce je dostatočná základná funkčnosť, stabilná verzia programu, dostatočná manuálová podpora, prehľadnosť používateľského prostredia, cenová dostupnosť, prípadne kompatibilita s operačným systémom Windows.

Konkrétne požiadavky sa budú ešte časom upravovať, no už teraz je zrejmé, že budeme potrebovať nástroj, ktorý bude obsahovať grafické používateľské rozhranie. Na pohodlné modelovanie, rýchly návrh a jednoduché úpravy návrhu by textový mód prinášal zbytočné komplikácie. Drvivá väčšina dnes dostupných nástrojov už grafický editor v sebe zahŕňa, takže s touto požiadavkou by nemal byť problém.

Podpora základných vlastností Petriho sietí ako je používanie miest a prechodov, ich animácia pri spustení simulácie musí byť zabezpečená. Jedná sa o základné vlastnosti, z ktorých náš model bude vychádzať.

Farebné Petriho siete v dnešnej dobe nie sú programovo dostatočne dobre zastúpené, takže táto funkcionálna nie je pre nás kľúčová. Samozrejme niektoré nové verzie programov túto možnosť ponúkajú, ale na úkor stability softvéru. To je v rozpore s našimi hlavnými kritériami, čiže túto možnosť uvítame len v prípade stabilnej verzie programu.

Časové Petriho siete poskytujú rozsiahlejšie možnosti modelovania a špecifickejšie nastavenia pri práci s časom. V našom prípade sa jedná o veľmi vhodnú funkciu, avšak taktiež nepatrí medzi hlavné kritériá. V budúcom riešení projektu sa možno prikloníme k programom s podporou časových Petriho sietí, avšak v tomto momente to prijímame len ako potenciálne využiteľnú funkčnosť navyše.

Hľadanie chýb v návrhu siete, resp. v jej simulácii poskytujú väčšinou len podnikové riešenia, preto na túto možnosť nespoliehame a simulovanie chýb bude vytvárané v našej réžii.

Poslednou veľmi významnou črtou programu musí byť cena. Možné alternatívy pre našu prácu sú licencie freeware, OpenSource alebo akademická verzia programu prípadne demo.

7.1 Programové nástroje

Výber konkrétnych nástrojov spočíval v prehľadávaní internetu a špecializovaných stránok, ktoré sa venujú modelovaniu Petriho sietí [7]. Taktiež podstatné boli odporúčania našej vedúcej projektu Ing. Jany Flochovej, Phd. , ktorá má v danej problematike dlhoročné skúsenosti a pracovala s rôznymi nástrojmi tohto typu.

Po konzultáciách a porovnaní jednotlivých nástrojov sme do užšieho výberu vybrali štyri programy, ktoré spĺňajú nami vyššie spomínané požiadavky. CPN Tools udržiavaný univerzitou v Dánsku, Platform Independent Petri Net Editor 2 vytvorený na univerzite v Londýne, Netlab pochádzajúci z univerzity v Nemecku a HPSim, ktorý je vyvíjaný súkromne. Takže väčšinou sa jedná o akademické produkty, ktoré prešli dlhodobým vývojom.

7.1.1 CPN Tools (Coloured Petri Nets)

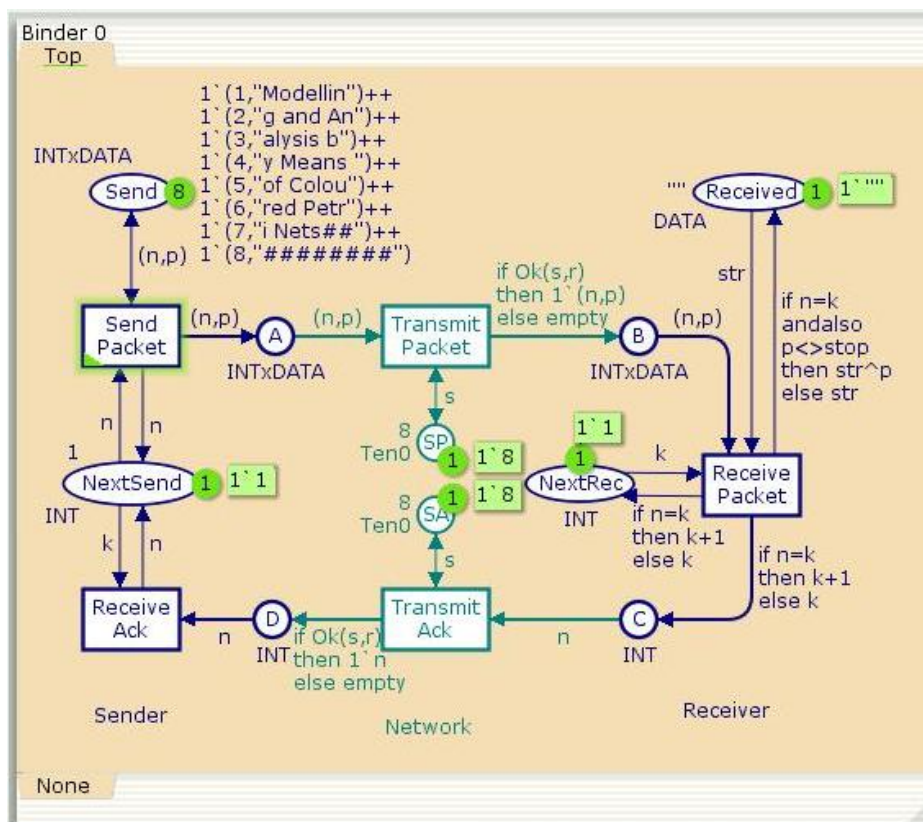
<http://wiki.daimi.au.dk/cpntools/cpntools.wiki>

Už z názvu vyplýva, že tento nástroj sa bude prevažne zameriavať na modelovanie, simuláciu a analýzu farebných Petriho sietí. Nahradil dlhé roky vyvíjaný CPN/Design Tool nástroj, s ktorým je kompatibilný.

Poskytuje prehľadné grafické editačné prostredie. Obsahuje analytické mechanizmy na zistenie základných parametrov siete ako je živosť, ohraničenosť, výkonnosť a pod. Samotná analýza sa rozdeľuje na analýzu simulácie a čiastkovú alebo úplnú analýzu miest a stavov. Taktiež kontrolné mechanizmy pri vytváraní siete ako je kontrola generovanej syntaxe a závislosti medzi elementmi.

Obsahuje podporu pre časované, nečasované ako aj hierarchické Petriho siete. Neobsahuje však podporu inhibítorov.

Simulátor má veľmi dobre spracované manuálové stránky s komplexným popisom funkcií programu spolu s názornými obrázkami. Dostupné sú aj rôzne príklady modelov vytvorených Petriho sietí.



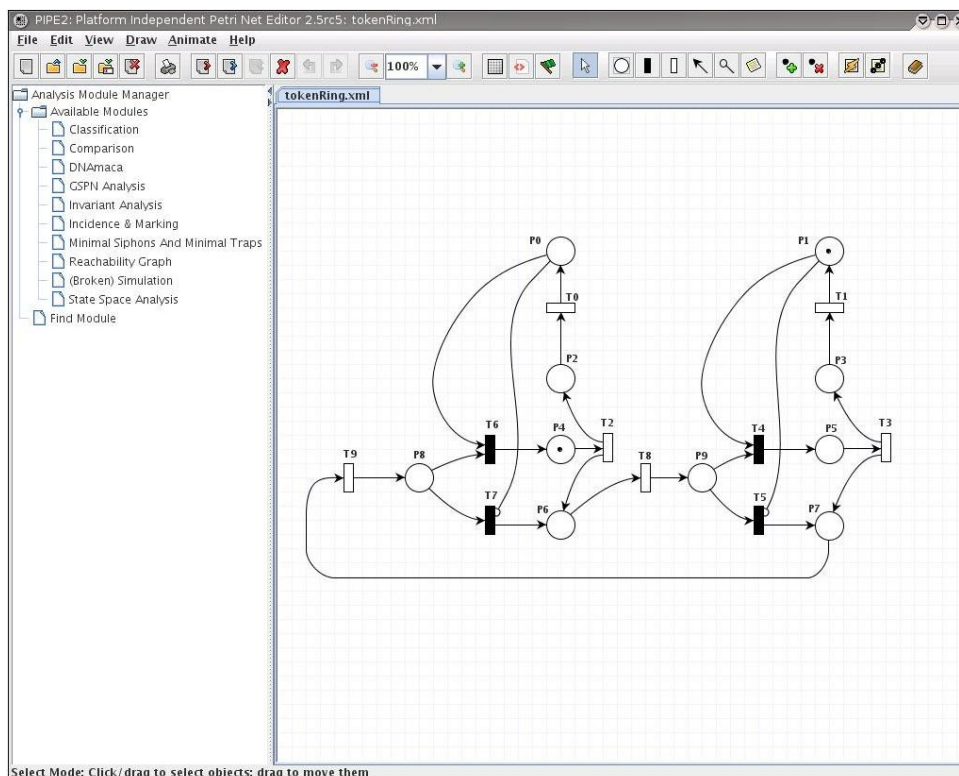
Obrázok č.1: Model jednoduchého protokolu v nástroji CPN Tools.

7.1.2 Platform Independent Petri Net Editor 2 (PIPE2)

<http://pipe2.sourceforge.net/>

Rýchly, výkonný a efektívny nástroj na modelovanie Petriho sietí vytvorený v prostredí multiplatformového jazyka Java. Poskytuje viaceré úrovne analýzy - analýza invariantov, analýza miest a stavov a analýza simulácie. Umožňuje tiež dotvoriť vlastné analytické moduly.

Jednoducho ovládateľné grafické rozhranie poskytuje rozsiahle možnosti nastavení, bezproblémové ovládanie a úpravu modelov. Animácia simulácie prehľadne znázorňuje časové správanie sa modelu. Obsahuje podporu pre stochastické Petriho siete. Generuje grafy dosiahnuteľnosti a incidenčné matice.



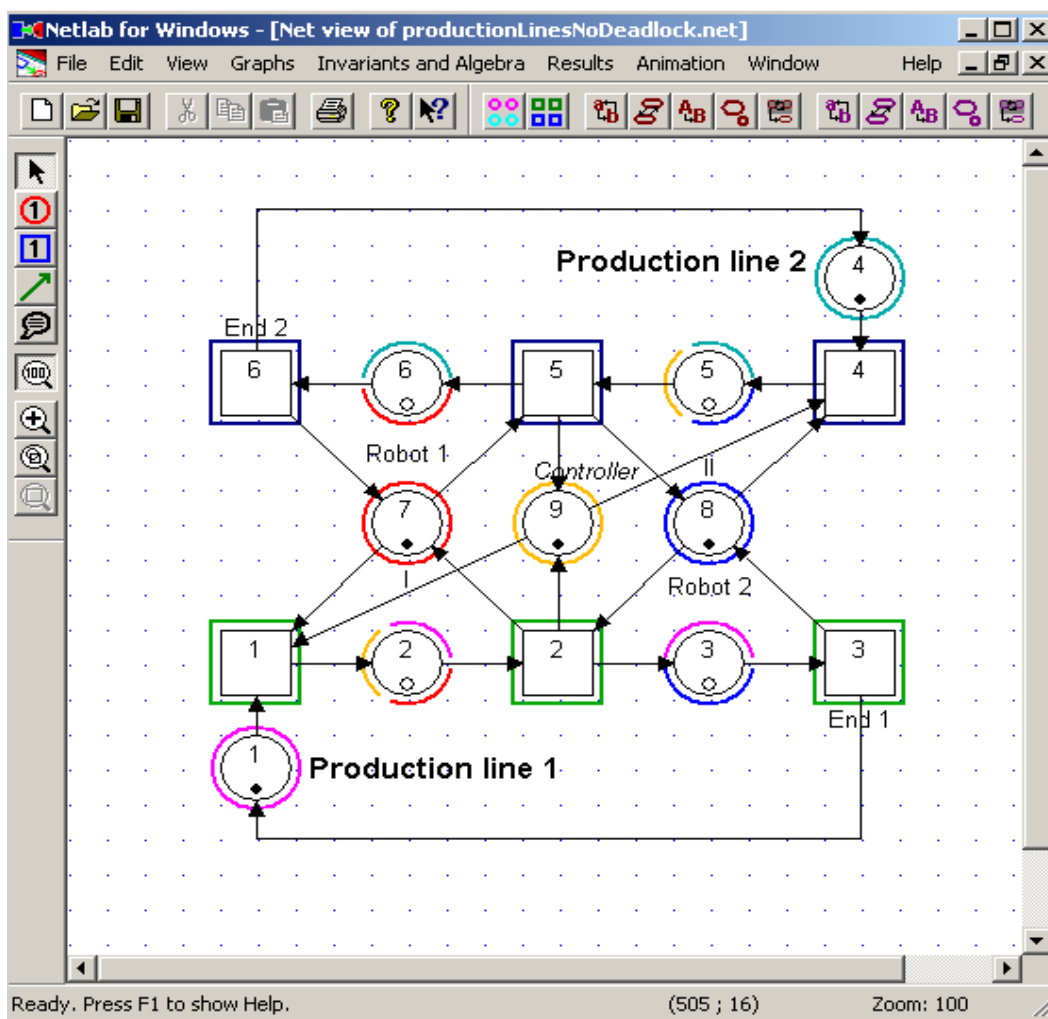
Obrázok č.2: Model token ring v programe PIPE2.

7.1.3 Netlab

<http://www.irt.rwth-aachen.de/en/downloads/petri-net-tool-netlab.html>

Simulačný nástroj na implementáciu Petriho sietí z dielne univerzity v Aachene. V manuálových stránkach uvádza možnosť diskretného modelovania stavového modelu Petriho siete nazývaného ako signalizačné Petriho siete. Podporuje prácu s invariantami miesta aj prechodu. Taktiež štruktúrovaná analýza umožňuje získavanie potrebných informácií o správaní sa siete, inhibítoroch a grafoch. Grafické prostredie a intuitívny editor vytvárajú jednoduchú cestu k návrhu simulačného modelu. Paralelné spracovanie viacerých modelov urýchľuje prácu a umožňuje sledovať vzájomné prepojenia sietí, resp. porovnávať výkonnosti jednotlivých modelov.

Za značnú nevýhodu považujeme to, že sa nám nepodarilo nájsť podrobné manuálové stránky a popis všetkých funkcií podporovaných týmto programom.



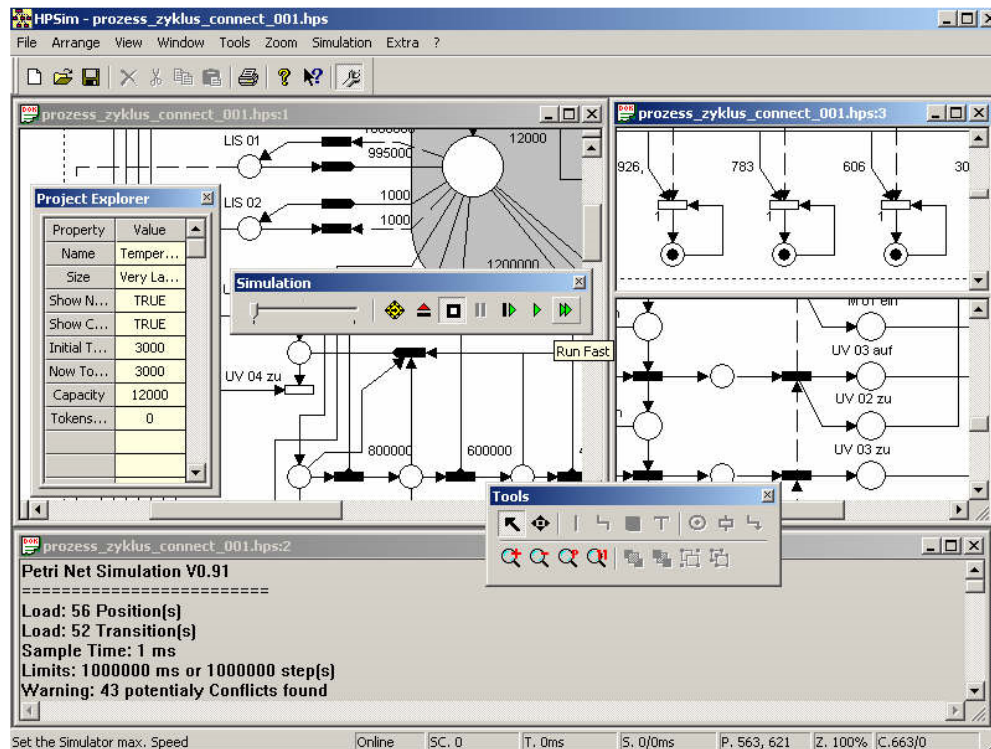
Obrázok č.3: Model produkčnej linky v programe Netlab.

7.1.4 HPSim

<http://www.winpesim.de/3.html>

Grafický editor Petriho sietí určený na vyučovacie účely, resp. pre užívateľov, ktorí sa chcú oboznámiť s funkcionalitou Petriho sietí. Umožňuje modelovanie stochastických aj časových sietí, váhové ohodnotenie hrán, časové prechody a pod. Grafický editor ponúka všetky štandardné funkcie na vytváranie a editáciu Petriho siete. Jednoduché rozhranie prináša komfort pre začiatočníkov aj pokročilých užívateľov.

Nevýhoda je nedostupnosť anglickej používateľskej príručky.



Obrázok č.4: Model systému v prostredí HPSim.

7.1.5 Zhodnotenie modelovacích nástrojov

Z vyššie analyzovaných nástrojov sme vybrali dva najvhodnejšie pre našu prácu. Prihliadali sme hlavne na funkcionality, efektívnosť ovládania a dostupnosť manuálových stránok o programe.

Do užšieho výberu sa dostali Platform Independent Petri Net Editor 2 a CPN Tools. Obidva poskytujú dostatočnú funkcionality a sú kvalitne zdokumentované. Avšak líšia sa v podpore pre určité vlastnosti Petriho sietí.

PIPE 2 umožňuje vytvárať modely stochastických sietí, aplikovať invarianty a rozsiahle možnosti analyzovania siete.

CPN Tools zase dokáže pracovať s farebnými aj časovými Petriho sieťami. Patrí medzi najpoužívanejšie prostriedky na modelovanie, poskytuje rozsiahlou podporou.

Ostáva teda zväziť, ktorý z dvoch vybraných nástrojov bude vhodný pre náš typ Petriho siete. Ak budeme potrebovať farebnú prípadne časovú Petriho sieť, odporúčaný nástroj je CPN Tools. Vzhľadom na komplexnosť protokolu OSPF budeme model implementovať v tomto nástroji.

Nasleduje prehľadová tabuľka s porovnaním jednotlivých nástrojov podľa [9].

	CPN Tools	Platform Independent Petri Net Editor 2	Netlab	HPSim
Licencia	Voľne dostupné	Voľne dostupné	Akademická	Voľne dostupné
Podpora typov Petriho sietí	Štandardné Časové	Štandardné	Štandardné Diskrétné	Štandardné Stochastiké Časové
Vlastnosti nástroja	Grafický editor Animácie Rýchla simulácia Stavy prechodov Analyzovanie Univerzálny výstup	Grafický editor Animácie Rýchla simulácia Stavy prechodov Zhustené stavy prechodov Univerzálny výstup Invarianty miest Invarianty prechodov Štruktúrálna analýza Analytické moduly	Grafický editor Animácie Rýchla simulácia Stavy prechodov Analyzovanie Zhustené stavy prechodov Univerzálny výstup Invarianty miest Invarianty prechodov	Grafický editor Animácie Rýchla simulácia Analyzovanie Univerzálny výstup
Platforma	PC, Linux PC, MS Windows	Java	PC, MS Windows	PC, MS Windows

Tabuľka č.1: Porovnanie simulačných nástrojov.

8 Protokol OSPF

Open Shortest Path First (OSPF) je protokol, ktorý bol vytvorený organizáciou IETF v rokoch 1988 až 1991. Vznikol za účelom zavedenia vysoko spoľahlivého, funkčného a od výrobcov nezávislého štandardu na smerovanie paketov IP protokolu. V súčasnosti je OSPF používaný ako smerovací protokol vnútri autonómnych systémov (AS), preto ho radíme medzi tzv. Interior Gateway protokoly (IGP). Prvá verzia OSPF bola popísaná v RFC 1131. Neskôr bola nahradená OSPF verziou 2 publikovanou v RFC 1247 a jej rozšíreniami (RFC 1583, 2178 a 2328). Najnovšia verzia OSPF v3 podporujúca štandard IPv6 je definovaná v RFC 2740 [10].

OSPF patrí medzi Link State smerovacie protokoly. Tie vytvárajú v pameti smerovača kompletnú mapu siete označovanú ako topologická databáza (taktiež známa pod názvom Link State Database – LSDB). Nad touto databázou sa vykonávajú výpočty pomocou algoritmu zvaného Shortest Path First (SPF), ktorý hľadá najvýhodnejšie cesty do oznamovaných sieťových destinácií. SPF používa Dijkstrov algoritmus, ktorý je z hľadiska procesorového výkonu výpočtovo náročný. Preto OSPF protokol obsahuje mechanizmy na zamedzenie príliš častého spúšťania prepočtu (napr. Hold time – minimálny čas medzi dvomi po sebe idúcimi kalkuláciami SPF).

8.1 Základná funkcionalita OSPF smerovačov

V tejto časti si popíšeme správanie smerovačov pracujúcich s protokolom OSPF [11].

- Smerovač vysiela v pravidelných intervaloch tzv. Hello pakety do priamo pripojených sietí a snaží sa nadviazať susedstvo s okolitými OSPF smerovačmi. V prípade, že sa smerovače zhodnú v požadovaných parametroch, stanú sa susedmi a naďalej si vymieňajú Hello pakety, čím udržujú svoje susedstvo aktívne.
- Aby mohlo dôjsť k výmene smerovacích informácií pomocou paketov Link State Advertisements (LSA), susedné smerovače musia nadviazať priateľstvo (tzv. adjacency). Smerovače dosiahnú finálny status priateľstva označovaný Full až keď majú zosynchronizované databázy LSDB.
- Smerovače si ukládajú prijaté LSA správy do LSDB a zároveň ich preposielajú svojim susedom. Týmto je zabezpečený konzistentný pohľad na topológiu siete pre každý OSPF smerovač.

- Keď topologická tabuľka (LSDB) smerovača obsahuje všetky potrebné informácie, spustí prepočet Dijkstrovho algoritmu SPF.
- Výsledkom prepočtu sú najkratšie a bezslučkové (loop-free) cesty do sieťových destinácií.

Ak v sieti nastane zmena (napr. linka sa stane nedostupnou), dotknutý smerovač pošle svojim susedom LSA správu o tejto zmene. Každý smerovač v sieti, ktorý obdržal túto informáciu musí spustiť SFP algoritmus a prepočítať novú cestu do danej siete, alebo sieť vyradiť zo smerovacej tabuľky, ak neexistuje náhradná cesta.

8.2 Typy sietí v OSPF

Protokol OSPF rozpoznáva niekoľko typov sietí:

- **Broadcast**

Ide o siete, ktoré dokážu prepojiť viac ako dva uzly v sieti a vyslané rámce môžu prijímať súčasne všetky z týchto uzlov. Táto vlastnosť značne ovplyvňuje spôsob vytváranie priateľstiev (adjacencies) medzi smerovačmi o čom si povieme neskôr. Príkladom broadcastových sietí sú Ethernet a FDDI.

- **Point-to-point**

Siete, ktoré prepájajú dva smerovače sériovou linkou. Smerovače sa stávajú vždy priateľmi (adjacent).

- **Non Broadcast Multi Access (NBMA)**

Sieť tohto typu dokáže prepojiť viac smerovačov, avšak nedokáže posilať broadcasty. Nie je teda možné, aby všetky smerovače prijali vyslanú správu. Príkladom takejto siete je Frame Relay.

- **Point-to-multipoint**

Ide o špeciálny prípad NBMA siete, v ktorej sú všetky linky chápané ako point-to-point lity.

8.3 Typy OSPF paketov

Protokol OSPF používa 5 rôznych typov správ na komunikáciu medzi smerovačmi. Tieto správy sú enkapsulované priamo do IP protokolu, pričom v hlavičke paketu je v poli IP Protocol

vždy uvedená hodnota 89 pre indikáciu OSPF protokolu. Jednotlivé typy správ sú popísané nižšie [12]:

- **Hello**

Hello pakety slúžia na objavovanie okolitých smerovačov pripojených na lokálnych linkách daného OSPF uzla. Slúžia na nadviazanie susedstva medzi smerovačmi a komunikáciu kľúčových parametrov. Sú posielané na multicastovú adresu 224.0.0.5 a obsahujú ID smerovača, ktorý pakety odoslal. V broadcastových sieťach sú Hello pakety posielané každých 10 s.

- **Database Description**

Tieto správy obsahujú popis topológie OSPF domény, teda prenášajú obsah Link State databáz medzi smerovačmi.

- **Link State Request**

Správy slúžiace na vyžiadanie si konkrétnych informácií z Link State databázy iného smerovača. Presne špecifikujú siete, pre ktoré žiadajú detailné smerovacie informácie.

- **Link State Update**

Tieto správy sú odpoveďou na Link State Requesty. Ich obsahom sú detailné informácie z LSDB databázy určené pre smerovač, ktorý si ich vyžiadal.

- **Link State Acknowledgement**

Správy slúžiace na overenie správneho doručenia ostatných OSPF správ. Explicitne potvrdzujú doručenie Link State Update paketov.

8.4 Typy správ Link State Advertisements

Každý Link State Update paket nesie tzv. Link State Advertisement (LSA). Základné typy LSA správ sú popísané nižšie [13]:

- **LSA typu 1 (Router LSA)**

Správa generovaná smerovačmi v danej oblasti na popísanie ich priamo pripojených liniek. Slúžia na prenos tzv. Intra-area smerovacích informácií. Tieto správy sa nešíria za hranice danej OSPF oblasti.

- **LSA typu 2 (Network LSA)**

Správa generovaná Designated Routerom (DR) broadcastovaj alebo NBMA siete na popis susedných smerovačov pripojených na danom segmente. Tieto správy sa nešíria za hranice danej OSPF oblasti.

- **LSA typu 3 (Summary LSA)**

Správa generovaná smerovačom na hranici OSPF oblastí na popis smerovacích informácií susedom mimo danej oblasti (ide o tzv. Inter-area routes).

- **LSA typu 4 (Summary LSA)**

Správa generovaná smerovačom na hranici OSPF oblastí na popis smerovacích informácií o tzv. ASBR (Autonomous System Boundary Router) smerovačoch pre susedov mimo danej OSPF oblasti.

- **LSA typu 5 (External LSA)**

Správa generovaná ASBR smerovačmi na popis smerovacích informácií redistribuovaných do OSPF oblasti.

8.5 Vytváranie susedstiev medzi OSPF smerovačmi

Ako už bolo spomenuté, susedstvo smerovačov nepostačuje na výmenu správ LSA. Preto smerovače musia nadviazať priateľstvo. Počas procesu nadväzovania priateľstva prechádzajú niekoľkými stavmi, ktorých význam je popísaný nižšie [14]:

- **Status Init**

V tomto stave si smerovače vymieňajú Hello pakety a vytvárajú susedstvo. Stav indikuje, že smerovač prijíma Hello pakety svojho suseda, no zatiaľ nebola vytvorená obojsmerná komunikácia. Tá sa vytvorí až vtedy, keď smerovače uvidia svoje vlastné ID v Hello paketoch prijatých od svojho suseda.

- **Status 2-way**

Ak smerovače vidia svoje ID v Hello paketoch svojho suseda, prejdú do stavu 2-way. V tomto bode sa smerovače rozhodnú, či uzatvoria priateľstvo. Na broadcastovej a NBMA sieti prejde smerovač do stavu Full iba so susedom zvaným Designated router (DR) a Backup Designated Router (BDR). S ostatnými susedmi zostane v stave 2-way. Na point-to-point a point-to-multipoint sieťach prejde smerovač do stavu Full so všetkými susedmi pripojenými na týchto linkách.

Poznámka: Prijatie paketu Database Descriptor (DBD) od suseda v Init stave taktiež spôsobí prechod do stavu 2-way.

- **Status Exstart**

Po zvolení DR a BDR ostatné smerovače vytvoria tzv. master-slave vzťah s DR a BDR, pričom master je smerovač s vyšším ID. Je dôležité poznamenať, že DR a BDR nemusia byť vždy v role mastera (aj keď logicky očakávame, že budú mať vyššie ID). Ich voľba za DR a BDR môže byť ovplyvnená aj manuálne nastavenou prioritou a ich ID môže byť pritom nižšie. Vo vzťahu master-slave potom hrajú rolu podriadeného slave.

- **Status Exchange**

V stave Exchange dochádza k výmene paketov Database Descriptor (DBD). Tie obsahujú iba LSA hlavičky a popis obsahu celej Link State databázy. Každý DBD paket má poradové číslo, ktoré môže pri prenose inkrementovať iba master. V tomto stave smerovače posielajú aj pakety Link State Request a Link State Update, ktoré obsahujú kompletnú LSA správu. Prijaté informácie sú porovnané s aktuálnou Link State databázou smerovača a ak sú v nej dostupné novšie informácie, databáza smerovača je podľa nich aktualizovaná.

- **Status Loading**

V tomto stave prebieha samotná výmena Link State databáz. Na základe súhrnných informácií z DBD paketov smerovače posielajú LSA Requesty a žiadajú v nich konkrétne informácie z Link State databázy spriateľného smerovača. DR a BDR na ne odpovedajú paketmi LSA Update. Tieto pakety sú prijímateľom potvrdzované (acknowledged).

- **Status Full**

Keď sú smerovače v stave Full, sú plne spriateľené a ich LSDB databázy sú zosynchronizované. Ide o bežný stav OSPF smerovačov. V prípade, že zostali zaseknuté v inom stave, znamená to problém s vytváraním priateľstiev. Výnimkou je stav 2-way, ktorý je bežný na broadcastových a NBMA sieťach. V prípade týchto sietí smerovač vytvára Full priateľstvá iba s DR a BDR smerovačmi. Ostatné smerovače vidí v stave 2-way.

- **Status Down**

V stave Down bolo priateľstvo so susedným smerovačom prerušené (napr. kvôli nedoručeniu Hello paketov).

9 Špecifikácia a návrh riešenia

Cieľom nášho tímového projektu je oboznámiť sa s problematikou modelovania systémov pomocou Petriho sietí. Rozhodli sme sa modelovať správanie protokolu OSPF pomocou farebných Petriho sietí. Fungovanie protokolu OSPF je bližšie popísané v kapitole 8. V nasledujúcej kapitole pojednávame o vhodných možnostiach simulácie vybraných častí a mechanizmov, ktoré OSPF protokol používa (napr. jeho správanie pri výpadku linky, formovanie susedských vzťahov medzi smerovačmi a pod.). Výsledkom kapitoly by mali byť špecifikované požiadavky, ktoré by mal náš model spĺňať.

Pre dosiahnutie uvedeného cieľa boli potrebné nasledujúce kroky:

- oboznámiť sa so základnou terminológiou z oblasti modelovania a simulácie systémov s dôrazom na oblasť farebných Petriho sietí,
- vykonať analýzu potrebných oblastí sieťového protokolu OSPF.

Kroky, ktoré budú potrebné pre splnenie zadania:

- analyzovať a osvojiť si prácu so simulačným prostredím pre implementáciu farebnej Petriho siete,
- aplikovať koncepcie a techniky modelovania systémov použitím farebných Petriho sietí,
- navrhnuť pilotný simulačný model protokolu OSPF pomocou nadobudnutých poznatkov z analýzy prístupov k modelovaniu pomocou CPN,
- vytvoriť funkčný simulačný model použitím simulačného programového prostredia,
- analyzovať výsledky simulácie modelu.

9.1 Simulácia vybraných mechanizmov protokolu OSPF

9.1.1 Čo chceme simulovať

Našu prácu by sme chceli zamerať na modelovanie dvoch hlavných štádií konvergenencie siete s implementovaným OSPF protokolom, a to vznik susedstiev medzi OSPF smerovačmi a výmena DDP (Database Description Packets). Hlavným cieľom je odsimulovať stavy, cez ktoré smerovače prechádzajú počas formovania susedstiev. Keďže prevádzka protokolu OSPF pozostáva z množstva procesov a udalostí, budeme si spočiatku všímať a modelovať len tie najzákladnejšie z nich. Počas simulácie budeme sledovať zložitosť modelu a postupne pridávať nové súčasti. Čiastkové činnosti OSPF protokolu sa pokúsime detailnejšie rozobrať na samostatných modeloch.

Tvorba modelu je veľmi dobrý spôsob, ako analyzovať správanie sa navrhovaného protokolu ešte pred jeho samotnou implementáciou, prípadne odhaliť chybové stavy, ktoré môžu nastať počas jeho prevádzky v budúcnosti. Modelovanie je taktiež užitočné pri detailnom oboznámovaní sa s činnosťou modelovaného protokolu.

9.1.2 Ako chceme simulovať

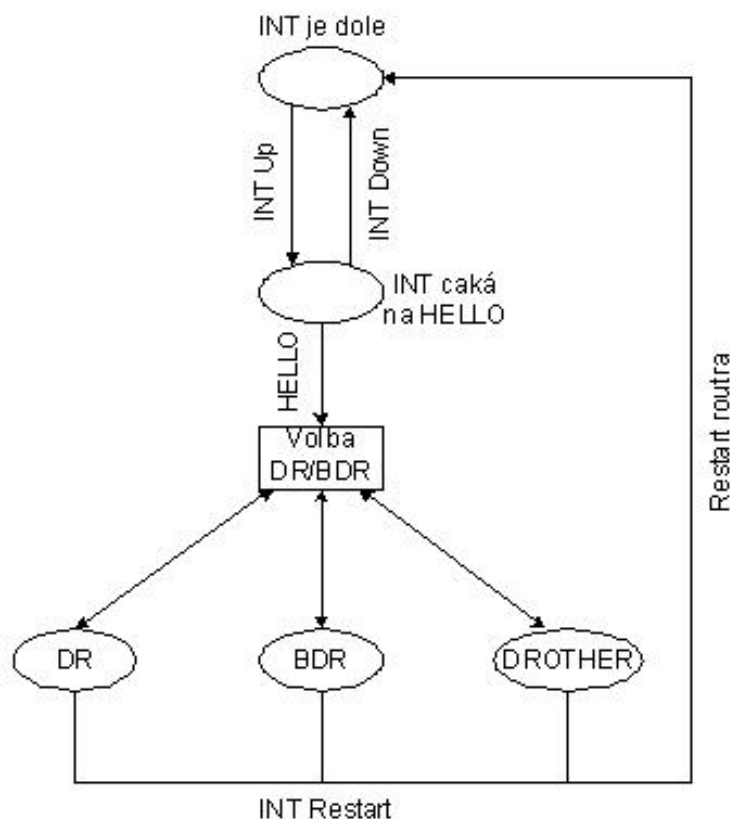
Ako typ modelu sme si zvolili Petriho siete, nakoľko sú vhodným matematickým nástrojom pre modelovanie a analýzu komplexných distribuovaných systémov so synchronizáciou udalostí. Keďže prevádzka OSPF protokolu v počítačovej sieti pozostáva z množstva prelínajúcich sa dejov, rozhodli sme sa bližšie zamerať na farebné Petriho siete. Tie boli vytvorené najmä za účelom modelovania zložitých systémov pozostávajúcich z veľkého množstva komunikujúcich procesov. Tento typ Petriho sietí sme preto zvolili za najvhodnejší spôsob simulácie prevádzky OSPF protokolu.

Ako simulačné prostredie sme si vybrali nástroj Coloured Petri Net Editor (CPN, pozri kapitolu 7.1.1).

9.2 Konceptia modelu

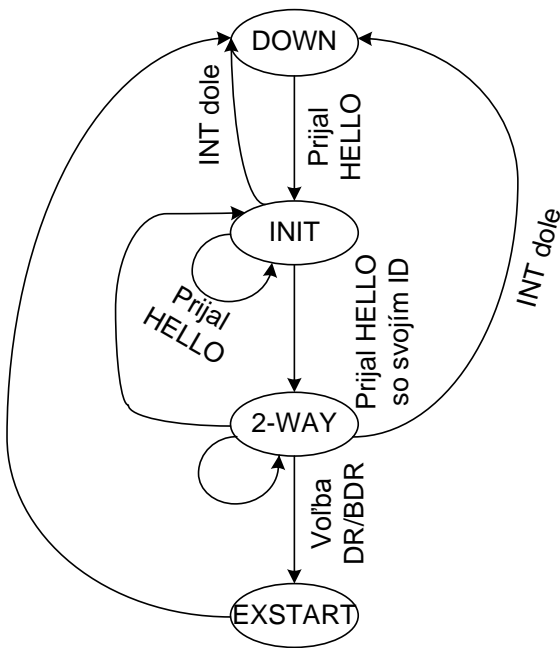
Nasledujúca kapitola popisuje základný návrh modelu protokolu OSPF. Pomocou stavových diagramov je opísaná konceptuálna funkcionálna požadovaných vlastností modelu a základná charakteristika jeho správania.

Počiatkové správanie sa protokolu OSPF je znázornené na obrázku č. 5. Obrázok znázorňuje nadviazanie susedstva medzi smerovačmi. Na začiatku je rozhranie smerovača vo vypnutom stave – stav „INT je dole“ (down state). V tomto stave si susedia nevymieňajú žiadne informácie. Po zapnutí rozhrania smerovač okrem vysielania vlastných paketov typu Hello čaká na Hello pakety od susedov. Po obdržaní a výmene Hello paketov medzi smerovačom a jeho susedmi smerovač iniciuje proces voľby DR a BDR.

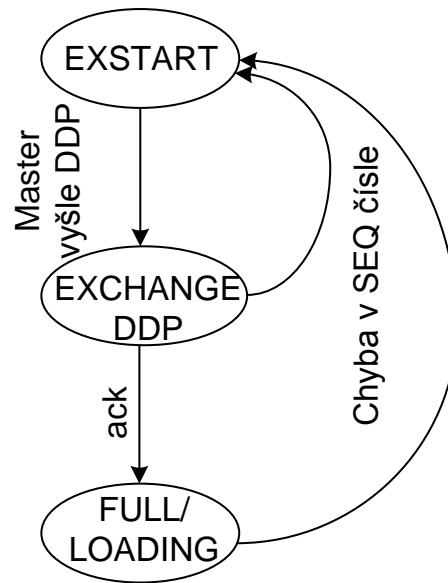


Obr. č.5: Stavový diagram procesu nadviazania spojenia a voľby DR, BDR.

Celkový model sme sa kvôli komplexnosti rozhodli rozdeliť na dva jednoduchšie moduly – modul inicializačnej fázy a modul výmeny DDP (pozri obrázky č. 6 a 7).



Obr. č.6: Modul inicializačnej fázy.



Obr.č.7: Modul Database Description Paketov

Modul inicializačnej fázy zobrazuje postupne stavy modelu smerovača vo fáze inicializácie. Nachádzame sa v prvom stave DOWN, čo znamená že rozhrania smerovača sú vypnuté a zatiaľ nebola získaná žiadna informácia. Po zapnutí rozhrania a prijatí HELLO paketu sa model dostane do stavu INIT. V tomto stave ale ešte nebola nadviazaná obojstranná komunikácia. Následne po prijatí HELLO paketu so svojím ID od susedného smerovača modul prechádza do stavu 2-WAY. Na konci tejto fázy po vzájomnej výmene informácií týkajúcich sa ID smerovačov je vykonaná voľba DR a BDR na základe typu siete (P2P alebo Broadcast/NBMA) a rozhodne sa o pokračovaní vytvárania susedstva. Model sa po voľbe DR/BDR dostane do stavu EXSTART. Smerovače sú pripravené na výmenu smerovacích informácií.

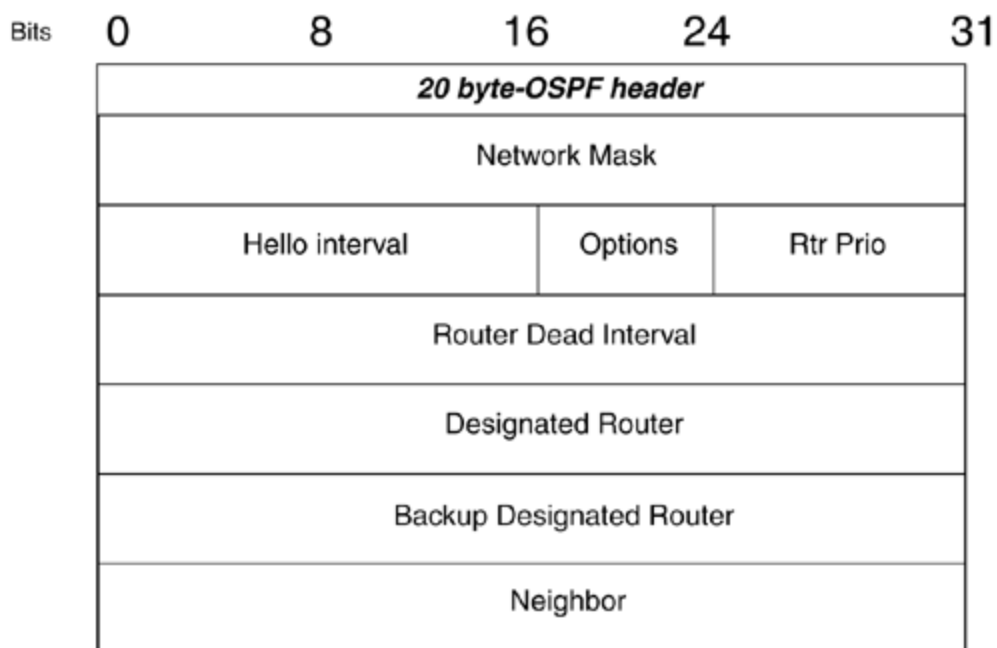
Počiatkový stav modulu DDP je EXSTART za predpokladu, že stav EXSTART je spustený iba medzi smerovačmi DROther a DR/BDR. Smerovače DR/BDR prechádzajú do stavu EXSTART za každých okolností. Cieľom stavu EXSTART je analyzovať vzťah master/slave medzi dvomi susednými smerovačmi. Rozhoduje sa na základe vyššej ID smerovača. Master vyšle ako prvý paket popisujúci smerovaciu databázu (DDP). Náš model teraz prejde do stavu EXCHANGE. Smerovače pošlú informáciu o svojej celej link-state database cez pakety DDP. Ak úspešne prebehne výmena smerovacích informácií, model sa dostáva do posledného stavu a to LOADING/FULL.

Stav **LOADING** umiestni každú informáciu, ktorá sa zdá byť nekompletná na link-state request zoznam. Každá vyslaná aktualizácia sa zaradí do zoznamu retransmission list, kde ostáva do potvrdenia doručenia. Stav **FULL** nastane, keď link-state request list neobsahuje žiadne informácie. Je určená príľahlosť, susedné smerovače sú úplne príľahlé a majú podobné databázy. Ak nastala chyba v sekvenčnom čísle (SEQ number), tak sa model vráti do stavu **EXSTART**. Túto fázu sme sa však kvôli jej celkovej zložitosti požadovaného modulu rozhodli zjednodušiť a spojiť do fázy **LOADING/FULL**.

9.3 Prototyp modelu

Ako už bolo spomínané v predchádzajúcich kapitolách, činnosť protokolu OSPF pozostáva z procesov a stavov, ktorými prechádzajú všetky smerovače v OSPF doméne. V tejto kapitole sa budeme venovať ich modelovaniu pomocou CPN.

Pre zjednodušenie výsledného modelu tak komplexného protokolu, akým je OSPF, sme sa rozhodli rozdeliť ho na čiastkové moduly. Tieto moduly sme sa kvôli prehľadnosti rozhodli rozčleniť na jednotlivé stavy alebo procesy. Modely týchto procesov si teraz opíšeme. Predpokladáme pritom, že každý OSPF smerovač si uchováva údaje o susedných smerovačoch v definovanej dátovej štruktúre. Tá bude obsahovať vybrané položky známe z hlavičiek OSPF paketov (pozri obrázok č.8), ako napr. Router ID, DR ID, BDR ID, NEIGHBOR ID susedných smerovačov, aktuálny status smerovača, príznaky (tzv. flags) a iné.



Obr. č.8: Polia v hlavičke Hello paketu.

V CPN sme si tieto štruktúry definovali takto:

color StavRoutra = record

*status: STATUS **

*RID: ID **

*DRID: ID **

*BDRID: ID **

*mBit: FLAG **

color HelloPkt = record

*RID: ID **

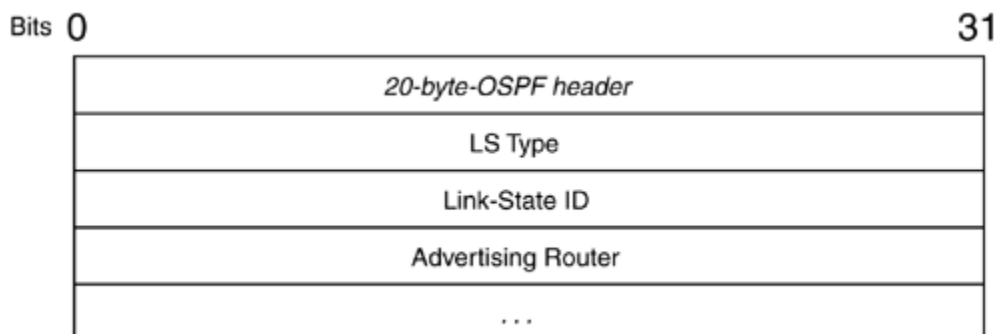
Štruktúra *StavRoutra* je použitá na zaznamenanie stavu smerovača. Jeho ID je uchovávané v premennej *RID* a identifikátory jeho DR a BDR sú zaznamenané v premenných *DRID* a *BDRID*. Smerovač prechádza jednotlivými OSPF stavmi, pričom aktuálny z nich je zaznamenaný v premennej *status*. Štruktúra *HelloPkt* nesie informácie o odoslanom Hello pakete. V pokračovaní našej práce tieto štruktúry doplníme o ďalšie premenné, ktoré budeme využívať a model urobíme detailnejším. Zatiaľ predpokladáme, že doručovanie OSPF správ prenášaných IP protokolom je spoľahlivé a bez strát.

Ďalšou štruktúrou je *DDPkt*, ktorá prenáša informácie o smerovaní v OSPF doméne. Táto štruktúra je zatiaľ zjednodušená vzhľadom na skutočnú hlavičku DDP paketu kvôli komplexnosti výmeny rôznych typov LSA paketov.

color DDPkt = record

*RID: ID **

*ARID: ID * (advertising RID)*



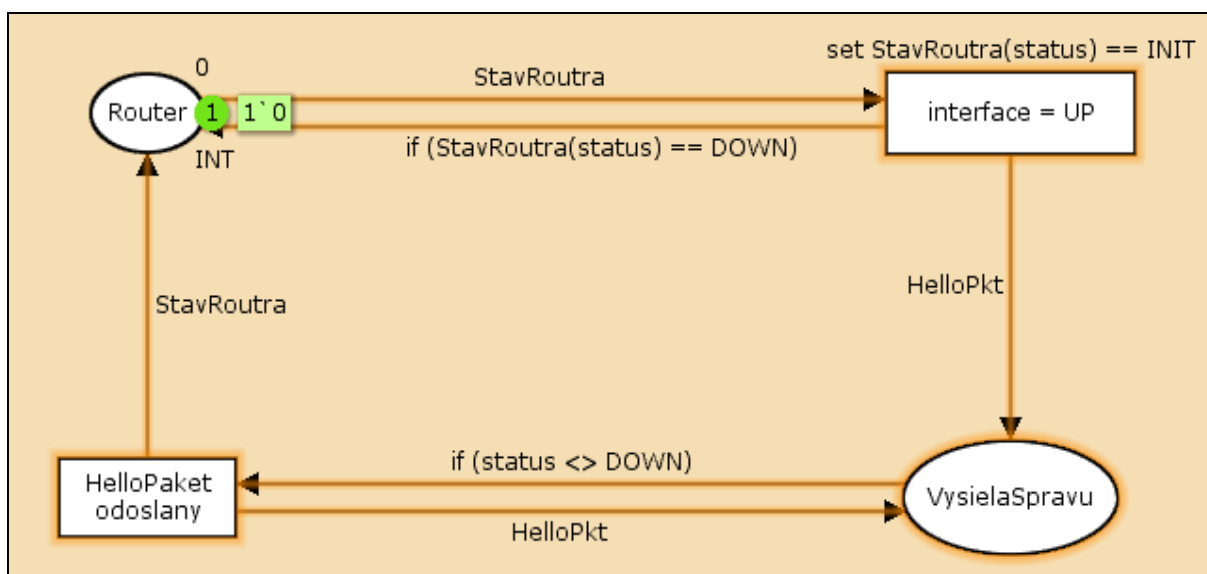
Obr. č.9: Polia hlavičky LSA paketu typu LSU.

9.3.1 Prechod zo stavu Down do stavu Init

Počiatkový stav OSPF smerovačov je Down. Počiatkové značkovanie modelu je v tomto stave nasledujúce:

```
1'StavRoutra({status=Dole, RID=10.0.0.1, DRID=Neznamy, DRID=Neznamy, BDRID=Neznamy, mBit=0}).
```

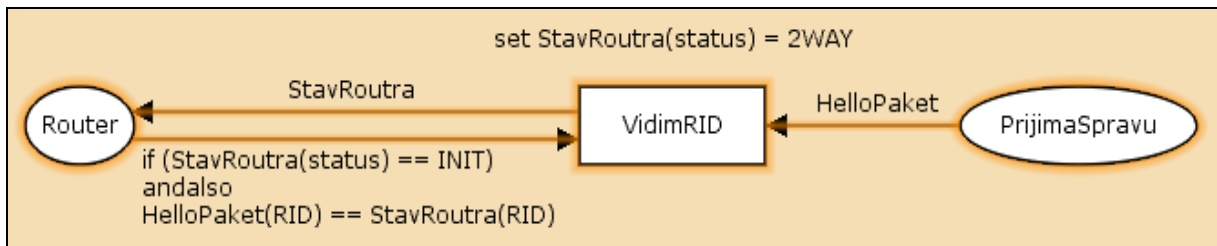
V tomto stave smerovač nevysiela a ani neprijíma pakety. V okamihu, keď jeho sieťové rozhranie príde do stavu UP (fyzicky pripojíme dátový kábel, alebo konfiguračne aktivujeme sieťové rozhranie), smerovač začne odosielať Hello pakety v pravidelných intervaloch a taktiež je schopný prijímať pakety od svojich susedov. Proces prechodu smerovača zo stavu Down do stavu Init je znázornený na obrázku č.9. Po spustení prechodu „*interface = UP*“ sa aktivuje rozhranie a status smerovača sa nastaví na INIT. Tým sa spustí proces vysielania vlastných Hello paketov za účelom oznamovania svojej prítomnosti v OSPF doméne (stav Vysiela spravu). Tým pádom je následne možné aktivovať prechod *HelloPaket odoslany*. Pri jeho spustení ale musí byť splnená podmienka, že sa status smerovača medzičasom nesmie vrátiť do stavu Down.



Obr. č.10: Model prechodu smerovača do stavu Init.

9.3.2 Prechod zo stavu Init do stavu 2Way

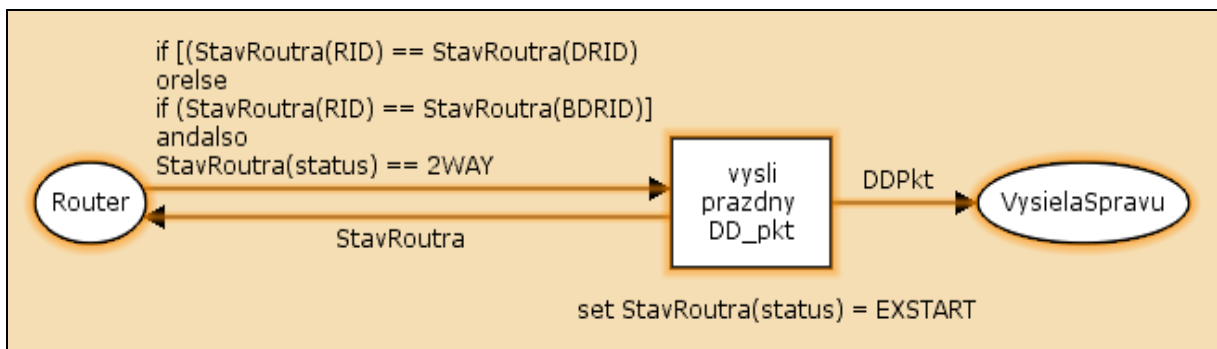
Na obrázku č.10 je znázornené prijatie Hello paketu od suseda, pričom sa skontroluje, či sa v prijatom Hello pakete nachádza ID prijímateľa (nášho smerovača). V prípade, že je splnená táto podmienka a smerovač je v stave INIT, prejde do stavu 2WAY. Tento stav značí, že výmena Hello paketov medzi smerovačmi je obojstranná a spätne potvrdená prítomnosťou ID prijímateľa v prijatom pakete.



Obr. č.11: Model prechodu smerovača do stavu 2 Way.

9.3.3 Prechod zo stavu 2 Way do stavu Exstart

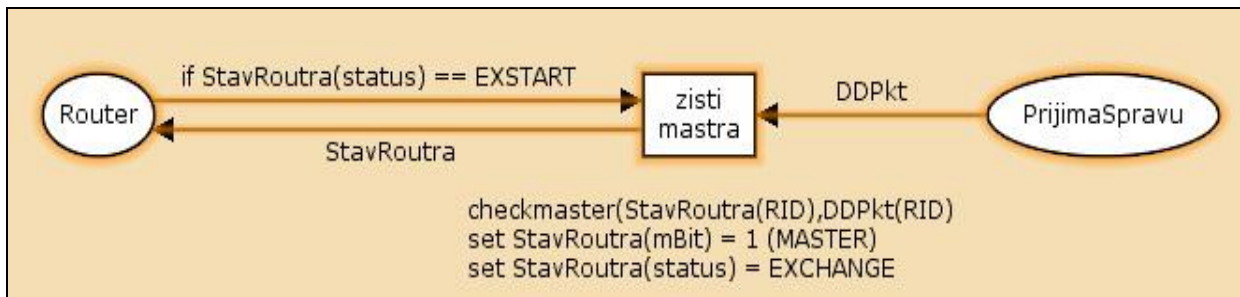
Prechod do stavu Exstart je možný len s tými smerovačmi, ktoré sú zvolené za DR a BDR. S ostatnými smerovačmi zostáva opisovaný smerovač v stave 2Way. Tento proces je znázornený na obrázku č.11.



Obr.č.12: Model prechodu smerovača do stavu Exstart.

9.3.4 Prechod zo stavu Exstart do stavu Exchange

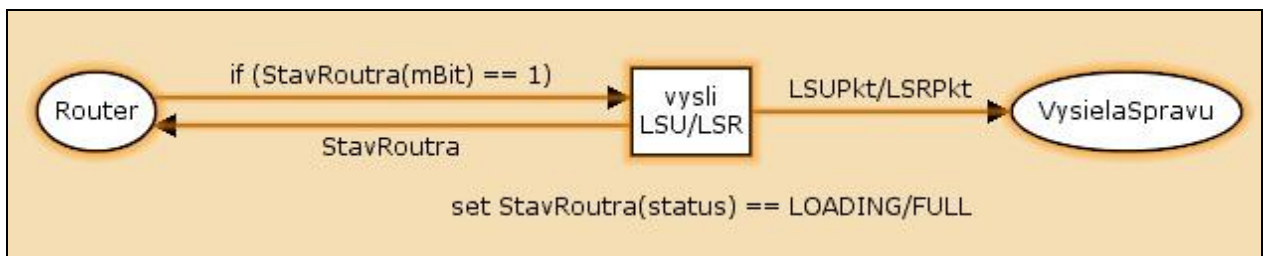
Na obrázku č.12 je znázornený proces prechodu smerovača do stavu Exchange. Tento proces zahŕňa aj voľbu tzv. Master a Slave roly pre smerovače, ktoré si idú vymieňať informácie. Smerovač s vyššou IP adresou bude Master, čo zabezpečíme nastavením príznaku mBit. Prechod do stavu Exchange znamená výmenu Database Description (DD) paketov.



Obr. č.13: Model prechodu smerovača do stavu Exchange.

9.3.5 Prechod zo stavu Exchange do stavu Loading/Full

Proces výmeny DD paketov obsahuje aj kontrolný mechanizmus, či je smerovač, ktorý vysiela správy v roli Master. Tento model je zatiaľ interpretovaný zjednodušene a na základe dôkladnejšej analýzy tohoto stavu sa rozhodneme o následnom detailnejšom rozpracovaní modelovania prechodu smerovača do stavu Loading/Full, vzhľadom na veľký počet rôznych typov vyžiadaných a odoslaných Link State Request/Update paketov (LSU/LSR).



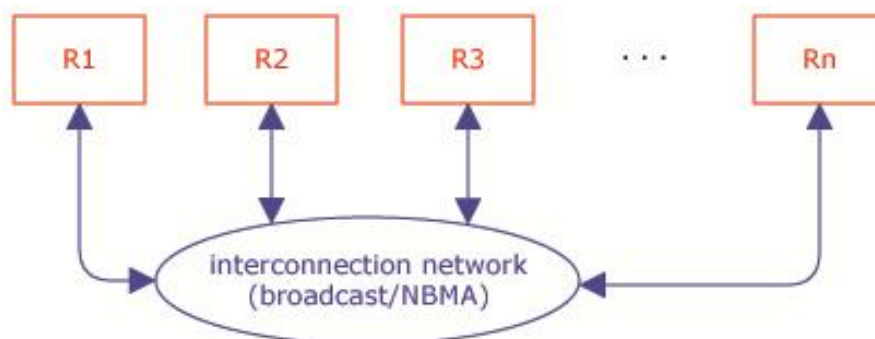
Obr. č.14: Model prechodu smerovača do stavu Loading/Full.

10 Implementácia modelu v nástroji CPN Tools

Samotná implementácia modelu v závere predmetu Tímový projekt II sa podstatne líši od návrhu modelu z predmetu Tímový projekt I. Po priebežnom študovaní funkcionality a syntaxe nástroja CPN Tools bola koncepcia modelu zmenená v takom rozsahu, aby umožňovala plnohodnotný opis protokolu OSPF a aby sa v čo najväčšej miere približovala špecifikácii štandardom RFC popisujúcim OSPF (pozri kapitolu 8). Prototyp z predmetu Tímový projekt I síce bol navrhovaný podľa RFC, ale nespĺňal podmienky kompatibility s nástrojom CPN Tools a po úvodnom pokuse implementácie sme boli nútení s cieľom čo najkorektnejšieho popisu zvoleného protokolu tento návrh počas semestra postupne meniť. Základným rozdielom oproti návrhu je oveľa detailnejšie spracovanie stavov INIT, 2WAY, voľba DR/BDR routrov (pozri kapitolu 8.5 a 9.3.1), modularity a stratovosti paketov na linke. Stav EXSTART, EXCHANGE a LOADING/FULL sú po zvolení dvojice routrov DR a BDR považované za implicitne vykonávané bez chybovosti a kvôli celkovej náročnosti modelu ani neboli implementované. Konkrétnejšie je funkcionality modelu popísaná v nasledujúcich kapitolách.

10.1 Koncepcia modelu siete s OSPF smerovačmi

Celková popisovaná štruktúra modelu je znázornená na *Obr. č.15*. Top-level hierarchia modelu pozostáva z modelu broadcastovej, prípadne NBMA siete ako entity prepájajúcej dve alebo viacero inštancií modelu smerovača so spusteným OSPF procesom. Entity $R1$ až Rn sú modulárne a do siete je možné zapojiť ľubovoľný počet týchto entít.



Obr. č.15: Top-level hierarchia modelu.

V nasledujúcej práci používame kvôli jednoduchosti a prehľadnosti celkového modelu tri inštancie entity smerovača prepojených broadcastovou sieťou.

10.2 Popis funkcionality

Implementovaný model opisuje správanie sa troch OSPF smerovačov v broadcastovej sieti, ich nadviazanie spojenia a konvergenciu do stavu zvolených DR a BDR smerovačov. Tento stav sme v našej práci považovali za koncový stav skonvergovanej siete.

10.2.1 Prijímanie a odosielanie paketov

Prijímanie a odosielanie paketov je možné len v tom prípade, keď je sieťové rozhranie smerovača v stave UP, v našom prípade teda `int_state("UP")`. Smerovač dokáže prijať vždy len jeden hello paket, a ten musí potom spracovať. Kým nie je tento paket spracovaný, nie je možné prijímať ďalšie hello pakety, avšak odosielanie takýchto paketov je možné neustále.

Po prijatí hello paketu a po jeho spracovaní sa IP adresa odosielateľa tohto paketu pridá do zoznamu susedov v hello paketoch, ktoré sú následne našim smerovačom odosielené. Ošetrovaný je aj prípad pridania vlastnej IP adresy do zoznamu, ak smerovač prijme svoj vlastný hello paket zo siete. Po úspešnom prijatí paketu zo siete sa určí, či sa jedná o nového suseda, alebo suseda, ktorý už existuje v `NBR DB`. Následne sa vytvorí záznam o susednom smerovači a určí sa jeden zo stavov OSPF suseda - stav INIT alebo 2-WAY. Ak smerovač nenájde svoju IP adresu (RID) v zozname susedov prijatého hello paketu, pridá suseda do stavu INIT, inak do stavu 2-WAY. Kým prijatý záznam nebude spracovaný a pridaný do `NBR DB`, zablokuje sa prijímanie ďalších hello paketov.

10.2.2 Pridávanie do NBR DB

Ak sused ešte neexistuje v `NBR DB`, tak je pridaný cez prechod `add new nbr`. Ak sa v `NBR DB` už nachádza, pridá sa cez prechod `treat old neighbor`. Ošetrované je aj viacnásobné pridanie smerovača do `NBR DB`. Stav existujúcich záznamov v `NBR DB` sa menia podľa toho, či sa smerovače obojsmerne vidia pomocou hello paketov (stavy INIT a 2-WAY).

10.2.3 Voľba DR a BDR

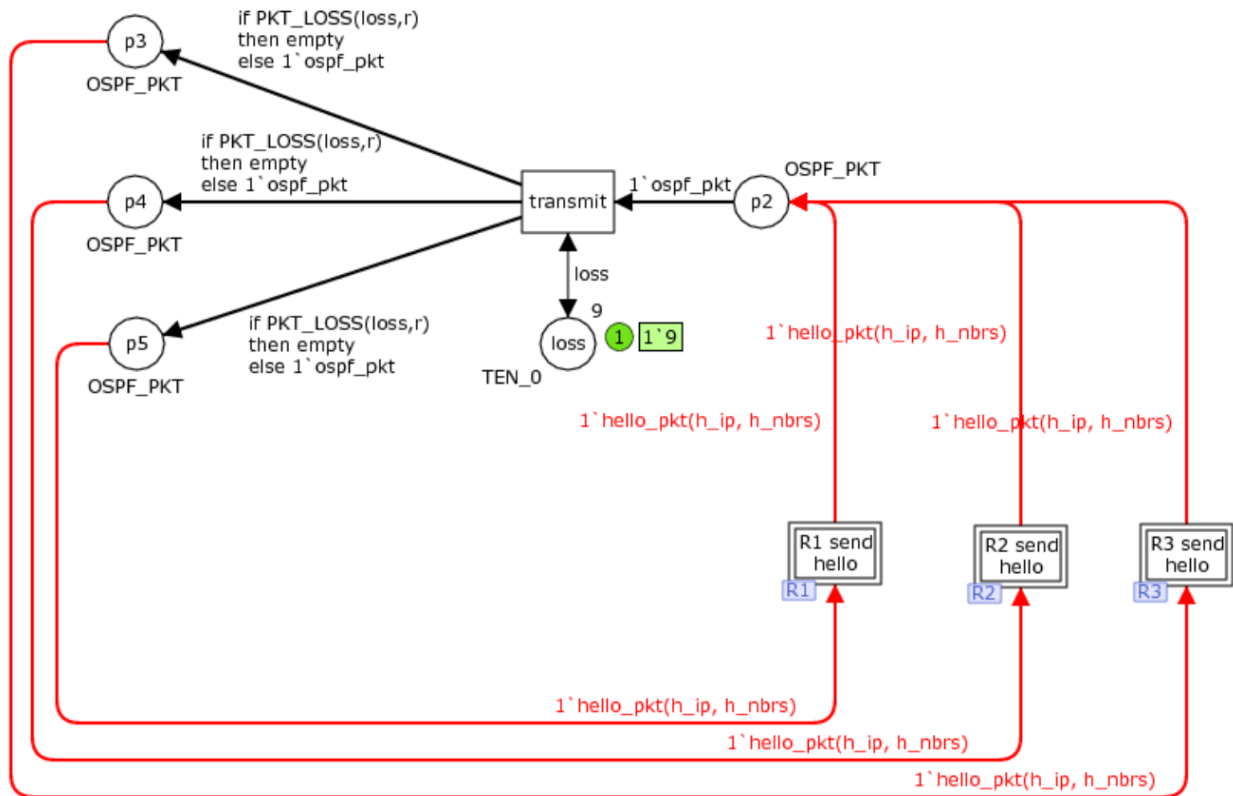
V sieti je za DR smerovač označený taký smerovač, ktorý má najvyššiu IP adresu zo všetkých komunikujúcich smerovačov. Označenie BDR zas nesie OSPF smerovač, ktorý má druhú najvyššiu IP adresu zo všetkých komunikujúcich smerovačov (v nasledujúcom texte sa pojmy „IP adresa smerovača“ a „ID smerovača (Router ID)“ obmieňajú a považujú sa za ekvivalentné).

Voľba DR a BDR sa vykonáva v prípade, že v NBR DB existuje aspoň jeden smerovač, ktorý je v stave 2-WAY. Do tejto voľby je taktiež zahrnutý aj samotný smerovač, na ktorom voľba prebieha.

Pri sieťovom rozhraní môže nastať situácia, že sa dostane do stavu DOWN (aj pri začiatku našej simulácie vychádzame zo stavu DOWN). Z pohľadu smerovača je padnutie sieťového rozhrania úplným prerušením komunikácie a preto si vymaže záznamy o každom susedovi z hello paketov, ktoré odosiela do siete. OSPF proces na smerovači zostane DOWN, pokiaľ nebudú odstránené všetky záznamy z databázy susedov NBR DB a záznamy o DR a BDR smerovačoch. Pri prechode sieťového rozhrania do stavu UP si smerovač predbežne nastaví svoju IP adresu RID ako DR. Avšak po pridaní susedných smerovačov do NBR DB prebehne voľba DR/BDR, kedy môže dôjsť k zmene tohto nastavenia.

10.3 Hierarchické moduly architektúry

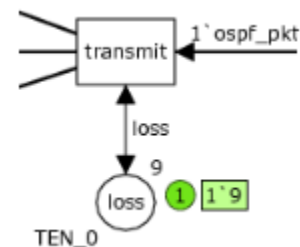
10.3.1 Modul prepájacej siete



Obr. č.15: Modul prepájacej siete.

Na obrázku *Obr. č.15* je znázornená sieť typu broadcast, ktorá prenáša prijaté pakety pomocou prechodu `transmit` do všetkých smerovačov v danej sieti (plní funkciu rozbočovača, prípadne prepínača v sieti). Prechod `transmit` je spustiteľný vždy, ak sa v mieste `p2` vyskytuje aspoň jedna značka (aspoň jeden paket je pripravený na prenos sieťou). Daný prechod značky (v našom prípade paketu) je zároveň ovplyvňovaný miestom `loss`, ktoré simuluje stratovosť paketov na linke takým spôsobom, že vyberá jedno náhodné číslo z intervalu 0 až 10 (premenná `loss`) a druhé číslo `r` viazané k náhodne zvolenému číslu z intervalu 1 až 10, ktoré následne navzájom porovná pomocou funkcie `PKT_LOSS`. Ak je číslo `loss` menšie ako `r`, paket sa prepošle linkou, ináč je simulovaná strata tohto paketu v sieti:

```
colset TEN_0 = int with 0..10;
colset TEN_1 = int with 1..10;
fun PKT_LOSS(loss:TEN_0, r:TEN_1) = (loss<=r);
```



Obr. č.16: Simulácia stratovosti.

Premenná `loss` vo funkcii `PKT_LOSS` vstupuje z miesta `loss` ako hodnota počiatočného značkovania, ktorú si používateľ môže sám zmeniť, a tým ovplyvniť percentuálnu stratu paketov v sieti. V našom modeli sme použili hodnotu 9, čo v praxi znamená 90 percentnú úspešnosť prenosu paketu sieťou. Pre potrebu inej percentuálnej úspešnosti je možné zmeniť hodnotu počiatočného značkovania a tým ovplyvniť stochastičnosť modelu.

Ostatné miesta ako `p2`, `p3`, `p4` a `p5` sú pomocné miesta, v ktorých sa kumulujú pakety pri prenose sieťou. Miesto `p2` je typu `OSPF_PKT` a slúži na zhromažďovanie všetkých paketov, ktoré čakajú na vstupe do siete a majú sa ňou prenášať. Miesta `p3`, `p4` a `p5` slúžia ako kumulatívny bod pre výstupné pakety. Dátový typ `OSPF_PKT`, ktorý obsahuje všetky možné typy paketov prechádzajúcich sieťou, je vlastne zjednotenie položiek `hello_pkt` typu `HELLO_PKT`, `dbd_pkt` typu `DBD_PKT`, `nbr_db` typu `NBR_DB`, `r_state` a `int_state` typu `STATE` definovaný nasledovne:

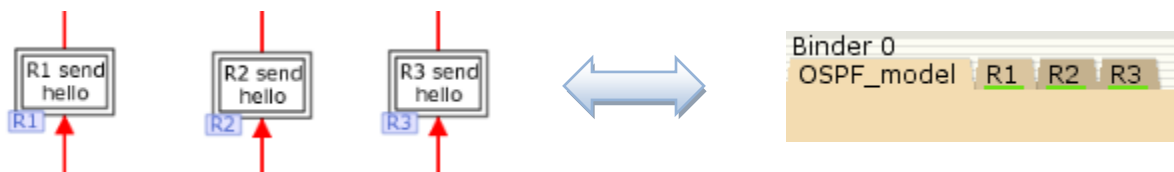
```
colset OSPF_PKT = union

    hello_pkt : HELLO_PKT +
    dbd_pkt   : DBD_PKT +
    nbr_db    : NBR_DB +
    r_state   : STATE +
    int_state : STATE;

colset HELLO_PKT = product IP * IP;
colset DBD_PKT   = product IP * MSG;
colset NBR_DB    = product IP * STATE;
colset STATE     = string;
colset MSG       = string;
colset IP        = string;
```

Na samotnom vstupe a výstupe do smerovačov pri procese nadväzovania spojenia je prenášaný paket s názvom `hello_pkt` dátového typu `HELLO_PKT` s atribútmi `h_ip` (skrátene z *hello_ip*, čo je premenná obsahujúca IP adresu smerovača odosielajúceho hello paket a `h_nbrs` (skrátene z *hello_neighbours*, ktorý obsahuje zoznam IP adres susedov daného smerovača odosielajúceho hello paket).

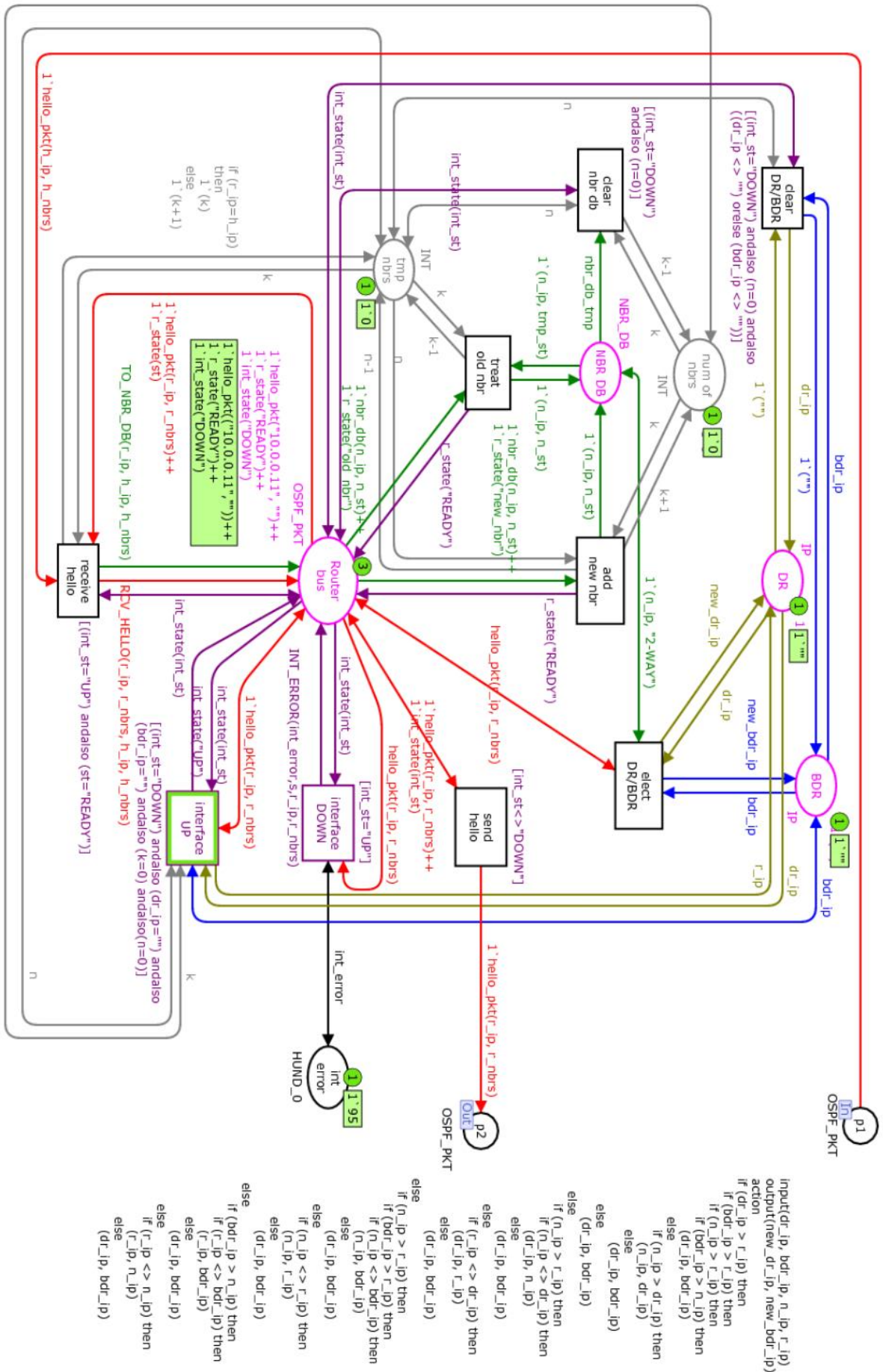
Podstránky znázornené na obrázku *Obr. č.17* reprezentujú napojenie jednotlivých inštancií modelu smerovača, ktorý si opíšeme v nasledujúcich kapitolách. Treba pripomenúť, že všetky smerovače sú inštanciami jedného modelu smerovača, majú rovnakú štruktúru a sú modulárne. Preto je možné pridať ľubovoľný počet smerovačov do siete a namapovať ich do vytvoreného modelu. Modré porty znázorňujú jednotlivé podstránky, ktoré si vymieňajú pakety s modelom siete a tým zabezpečujú hierarchický prístup podľa obrázka *Obr. č.15* v kapitole 10.1.



Obr. č.17: Tri inštancie modelu smerovača.

10.3.2 Modul smerovača

CPN model modulu smerovača je znázornený na obrázku *Obr. č.18*. Miesta `p1` a `p2` sú vstupné a výstupné porty komunikujúce s modelom siete z kapitoly 10.3. Pomocou týchto miest a pomocou rozhraní predstavovaných prechodmi `receive hello` a `send hello` smerovač prijíma pakety, spracováva ich, vyhodnocuje a vysiela naspäť do siete pre zabezpečenie vzájomnej komunikácie OSPF procesov smerovačov. Prechod `send hello` je pomocou guard-u `[int_st<>"DOWN"]` spustiteľný iba v prípade, ak je smerovač v aktívnom stave.



Obr. č.18: Modul smerovača v počítačnom značkovaní.

Centrálным bodom v modeli smerovača je miesto `Router bus`, ktoré z logického hľadiska tvorí akúsi internú zbernicu, ktorá slúži ako ústredný bod pre zhromažďovanie, uchovávanie a spracúvanie paketov a vnútorných premenných smerovača. Toto miesto je typu `OSPF_PKT` a uchováva si všetky svoje potrebné interné informácie, ako sú odosielaný typ hello paketu, aktuálny stav smerovača a stav jeho rozhraní. Pri komunikácii s ostatnými smerovačmi si smerovač tieto informácie aktualizuje a udržuje do momentu, kým nie je nejakým spôsobom odpojený od siete (napríklad vypnutím alebo náhodným padnutím rozhrania). Premenná `int_state` je typu `STATE` a uchováva si aktuálny stav sieťového rozhrania smerovača podľa.

Ďalšou vnútornou premennou v mieste `Router bus` je premenná `r_state`, ktorá je taktiež typu `STATE` a môže nadobúdať hodnoty `"READY"`, `"old_nbr"` alebo `"new_nbr"` v závislosti od toho, či je smerovač pripravený prijať nový paket zo siete, alebo práve spracúva prijatý paket.

Okrem spomenutých dvoch premenných si smerovač ešte uchováva informáciu o štruktúre aktuálneho hello paketu, ktorý odosiela do siete. Táto premenná má názov `hello_pkt` a je dátového typu `HELLO_PKT` s atribútmi vlastnej IP adresy a IP adres svojich už rozpoznaných susedov podľa hlavičky OSPF paketu (pozri *Obr. č.8*, kapitola 9.3).

Prislúchajúcimi prechodmi k miestu `Router bus` sú prechody `interface UP` a `interface DOWN`. Pri inicializácii smerovača je prechod `interface UP` jediným spustiteľným prechodom modelu. Po spustení tohto prechodu sa stav rozhrania sa zmení z hodnoty `"DOWN"` na hodnotu `"UP"`. Tým je smerovač pripravený odosielať hello pakety do siete a prijímať pakety od iných smerovačov. Smerovač si po spustení tohto prechodu dočasne sa nastaví hodnotu DR na hodnotu vlastnej IP adresy (vlastný generický inicializačný krok potrebný k následnej voľbe DR/BDR). Prechod `interface DOWN` slúži na vypnutie rozhrania a prechod smerovača do stavu DOWN. Na tento prechod sme taktiež aplikovali stochastickú vlastnosť náhodného padnutia rozhrania `int_error`, ktorá s pravdepodobnosťou 5% zhodí rozhranie počas fungovania simulácie. Zvolená pravdepodobnosť je vzhľadom k reálnym smerovačom prehnaná, je však takto zvolená schválne kvôli viditeľnosti a overeniu správnosti fungovania správania sa smerovača v priebehu simulácie. Po vykonaní prechodu `interface DOWN` sa vynulujú všetky hodnoty uchovávanej vnútornej databázy susedov NBR DB, a hodnoty DR a BDR smerovačov. K ich opätovnému naplneniu dôjde až po obnovení komunikácie so okolitými smerovačmi.

Ďalšie miesto uchovávajúce stavové premenné je `NBR DB`, ktoré zaznamenáva databázu všetkých susedných OSPF smerovačov. Prislúchajúce prechody, cez ktoré sa značky do tohto miesta dostávajú, sú `add new nbr` a `treat old nbr`. Tieto prechody vytvárajú, prípadne aktualizujú tabuľku OSPF susedov v mieste `NBR DB` podľa analýzy zdrojovej IP adresy smerovača, od ktorého paket prišiel a aktuálneho uchovaného OSPF stavu tohto smerovača. Ak

už náš smerovač obsahuje záznam IP adresy smerovača od ktorého prijal OSPF hello paket, spustiteľným sa stane prechod `treat old nbr`. Naopak ak smerovač neobsahuje záznam prijatej IP adresy, spustiteľným bude prechod `add new nbr` a tabuľka v mieste `NBR DB` sa aktualizuje o nový záznam OSPF suseda.

Miesta DR a BDR udržujú IP adresu aktuálneho DR smerovača a náhradného, BDR smerovača v broadcastovej sieti. Výskyt týchto adries v miestach DR/BDR riadi prechod `elect DR/BDR`, ktorý navzájom porovnáva prijaté IP adresy so svojou a na základe vyhodnotení určí najvyššiu IP adresu pre DR a druhú najvyššiu pre BDR smerovač. Kódový segment ktorý zisťuje spomínané okolnosti je definovaný nasledujúco:

```
input(dr_ip, bdr_ip, n_ip, r_ip)
output(new_dr_ip, new_bdr_ip)
action if (dr_ip > r_ip) then
    if (bdr_ip > r_ip) then
        if (n_ip > r_ip) then
            if (bdr_ip > n_ip) then
                (dr_ip, bdr_ip)
            else
                if (n_ip > dr_ip) then
                    (n_ip, dr_ip)
                else
                    (dr_ip, bdr_ip)
        else
            (dr_ip, bdr_ip)
    else
        if (n_ip > r_ip) then
            if (n_ip <> dr_ip) then
                (dr_ip, n_ip)
            else
                (dr_ip, bdr_ip)
        else
            if (r_ip <> dr_ip) then
                (dr_ip, r_ip)
            else
                (dr_ip, bdr_ip)
else
    if (n_ip > r_ip) then
        if (bdr_ip > r_ip) then
            if (n_ip <> bdr_ip) then
                (n_ip, bdr_ip)
            else
                (dr_ip, bdr_ip)
        else
            if (n_ip <> r_ip) then
```

```
        (n_ip, r_ip)
    else
        (dr_ip, bdr_ip)
else
    if (bdr_ip > n_ip) then
        if (r_ip <> bdr_ip) then
            (r_ip, bdr_ip)
        else
            (dr_ip, bdr_ip)
    else
        if (r_ip <> n_ip) then
            (r_ip, n_ip)
        else
            (dr_ip, bdr_ip)
```

Podmienkou spustiteľnosti prechodu `elect DR/BDR` je existencia minimálne jedného suseda v tabuľke miesta `NBR DB`, ktorý je s našim smerovačom vo vzťahu 2-WAY.

Miesta `tmp nbrs` (počet susedov, ktorí sa práve spracúvajú na zbernici smerovača) a `num of nbrs` (počet susedov pridaných do `NBR DB`) sú pomocné miesta, ktoré slúžia na udržiavanie vnútorných premenných potrebných pre internú správu, ako napríklad pri ukladaní nových položiek do tabuľky susedov, prípadne mazaní tabuľky DR/BDR pri inicializácii alebo vypnutí smerovača. Tieto hodnoty sú využívané prechodmi `clear DR/BDR` a `clear nbr db` pri nulovaní záznamov v DR/BDR a v databáze susedov.

10.4 Použité štruktúry a premenné

Pre úplnosť uvádzame kompletný zoznam použitých premenných a štruktúr využívaných v modeli smerovača:

```

colset INT = int;
colset BOOL = bool;
colset STRING = string;
colset STATE = string;
colset MSG = string;
colset IP = string;
colset TEN_0 = int with 0..10;
colset TEN_1 = int with 1..10;
colset HUND_0 = int with 0..100;
colset HUND_1 = int with 1..100;
colset NBR_DB = product IP * STATE;
colset HELLO_PKT = product IP * IP;

colset OSPF_PKT = union
  hello_pkt : HELLO_PKT +
  nbr_db : NBR_DB +
  r_state : STATE +
  int_state : STATE;

var ospf_pkt: OSPF_PKT;
var nbr_db_tmp: NBR_DB;
var st: STATE;
var int_st: STATE;
var r_nbrs :IP;
var r_ip :IP;
var n_ip: IP;
var n_st: STATE;
var dr_ip: IP;
var bdr_ip: IP;
var new_dr_ip: IP;
var new_bdr_ip: IP;
var tmp_st: STATE;
var tmp_st2: STATE;
var h_ip: IP;
var h_ip_g: IP;
var h_nbrs :IP;
var loss: TEN_0;
var int_error: HUND_0;
var s: HUND_1;

```

```
var r: TEN_1;
var k: INT;
var n: INT;

fun PKT_LOSS(loss:TEN_0, r:TEN_1) = (loss<=r);

fun INT_ERROR(prob:HUND_0, x:HUND_1, r_ip:IP, r_nbrs:IP) =
if (prob<=x) then
  1`int_state("DOWN")++
  1`hello_pkt(r_ip, "")
else
  1`int_state("UP")++
  1`hello_pkt(r_ip, r_nbrs);
```

Funkcia zabezpečujúca prijatie a spracovanie hello paketu zo siete. Taktiež zabezpečuje editáciu hlavičky hello paketu - do položky `self_nbrs` pridá suseda, od ktorého smerovač práve obdržal hello paket (ak sa daný smerovač už v tomto zozname nenachádza). Funkcia taktiež rozlišuje, či je prijatý hello paket od smerovača, ktorý už existuje v databáze susedov `NBR DB`, alebo či ide o nový smerovač:

```
fun RCV_HELLO(self_ip:IP, self_nbrs:IP, r_ip:IP, r_nbrs:IP) =
if ((isSubstring r_ip self_nbrs) orelse (self_ip = r_ip)) then
  if (self_ip = r_ip) then
    1`hello_pkt(self_ip, self_nbrs)++
    1`r_state("READY")
  else
    1`hello_pkt(self_ip, self_nbrs)++
    1`r_state("old_nbr")
else
  1`hello_pkt(self_ip, self_nbrs^r_ip^", ")++
  1`r_state("new_nbr");
```

Funkcia zabezpečuje vytvorenie záznamu o susednom smerovači na zbernici príjemcu a prípravu na jeho pridanie do `NBR DB`:

```
fun TO_NBR_DB(self_ip:IP, r_ip:IP, r_nbrs:IP) =
if (self_ip <> r_ip) then
  if (isSubstring self_ip r_nbrs) then
    1`nbr_db(r_ip, "2-WAY")
  else
    1`nbr_db(r_ip, "INIT")
else empty;
```

10.5 Počiatočné značkovanie siete

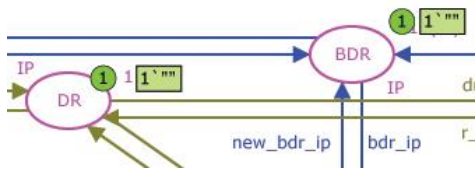
Počiatočné značkovanie siete je znázornené na obrázku *Obr. č.18* zelenými obdĺžnikmi pri danom mieste.

Počiatočné značkovanie smerovača:

```
1`hello_pkt("10.0.0.11", "")++
1`r_state("READY")++
1`int_state("DOWN")
1`hello_pkt("10.0.0.11", "")++
1`r_state("READY")++
1`int_state("DOWN")
```

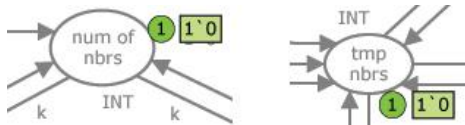
- `hello_pkt` nastavený iba s hodnotou vlastnej IP adresy, prázdny zoznam susedov
- `r_state` hodnota nastavená na `READY`, smerovač je pripravený na spustenie
- `int_state` nastavený na `"DOWN"`, čo je inicializačný stav rozhrania smerovača pri zapnutí

Počiatočné značkovanie miest DR a BDR:



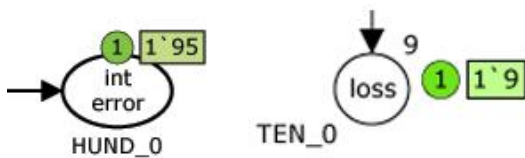
- DR – dátový typ IP s hodnotou prázdneho reťazca
- BDR – dátový typ IP s hodnotou prázdneho reťazca

Počiatočné značkovanie miest vnútorných premenných:



- inicializačné hodnoty rovné 0

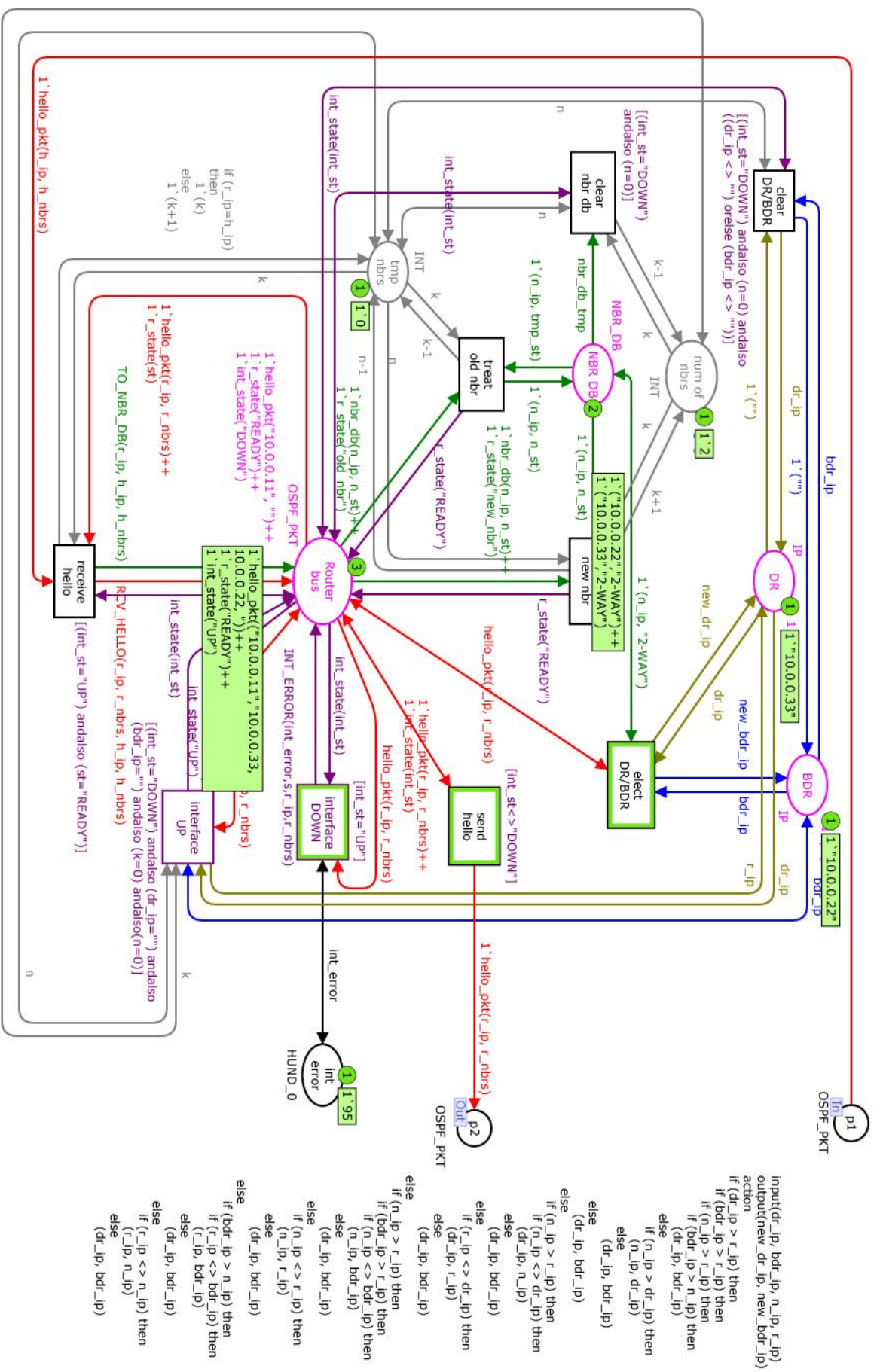
Počiatočné značkovanie miest určujúcich pravdepodobnosti:



- `int_error` parameter rovný 95 (5 percentná šanca zlyhania rozhrania), možnosť vlastnej úpravy
- `loss` parameter rovný 9 (10 percentná stratovosť), možnosť vlastnej úpravy

10.6 Koncové značkovanie (skonvergovaná sieť)

Koncové značkovanie siete je znázornené na obrázku *Obr. č.19* zelenými obdĺžnikmi pri danom mieste. Skonvergovaná sieť s tromi smerovačmi má v mieste `NBR_DB` dve IP adresy susedov so stavmi 2-WAY, v mieste `num of nbrs` má počet susedov a v miestach `DR` a `BDR` sú IP adresy DR a BDR smerovačov.



Obr. č.19: Modul smerovača v koncovom značkovaní.

11 Overovanie funkčnosti v prostredí CPN Tools

11.1 Simulácia modelu

Simulácia modelu je vykonávaná pomocou nástroja Sim v prostredí CPN Tools (pozri *Obr. č.20*). Pomocou prvého tlačidla zľava je možné inicializovať sieť do počiatočného značkovania ak tomu tak nie je pri samotnom otvorení a načítaní siete do programu CPN Tools. Druhé tlačidlo zľava zastavuje simuláciu spustenú tlačidlom *play* (tretie sprava). Číselná hodnota v tomto tlačidle udáva, koľko krokov (spustení prechodov) CPN Tools vykoná pred automatickým zastavením. Táto funkcia je v našom prípade z časového hľadiska dosť nepraktická. Naopak často využívanou funkciou je krokovanie simulácie (tlačidlo v strede), ktorá umožňuje používateľovi spúšťať jednotlivé prechody kliknutím na ním zvolený prechod. Taktiež užitočnou funkciou je funkcia okamžitého vykonania konkrétneho počtu prechodov, ktorá je aktivovaná druhým tlačidlom sprava.



Obr. č.20: Ovládací panel simulácie.

Pre komplexnosť a jednoznačnosť procesu simulácie sme si zadefinovali v modeli tzv. *Monitor*, ktorý zastaví simuláciu v momente, keď je splnený zadefinovaný predikát (v našom prípade je predikátom koncové značkovanie siete). Dôvod vloženia tohto monitora je to, že náš model je neohraničená sieť, ktorá sa môže vykonávať donekonečna. Nás však pri simulácii väčšinou zaujíma to, či daná sieť skonvergovala do ustáleného stavu. V tomto prípade je použitie monitora veľmi užitočné. Predikát monitora je definovaný nasledujúco:

```
fun pred (
    OSPF_model'BDR_1_mark : IP ms,
    OSPF_model'DR_1_mark : IP ms,
    OSPF_model'NBR_DB_1_mark : NBR_DB ms,
    OSPF_model'R2_BDR_1_mark : IP ms,
    OSPF_model'R2_DR_1_mark : IP ms,
    OSPF_model'R2_NBR_DB_1_mark : NBR_DB ms,
    OSPF_model'R3_BDR_1_mark : IP ms,
    OSPF_model'R3_DR_1_mark : IP ms,
    OSPF_model'R3_NBR_DB_1_mark : NBR_DB ms) =
    OSPF_model'BDR_1_mark <> 1`("")
    andalso
```

```

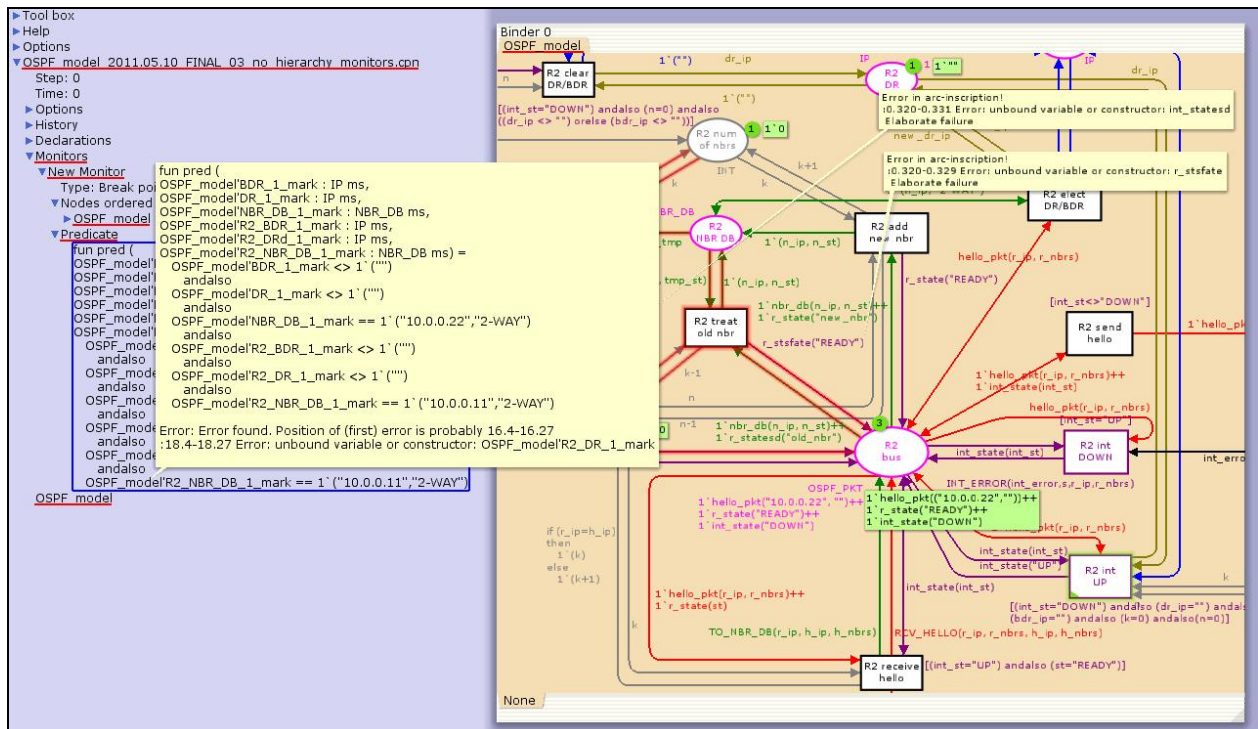
OSPF_model'DR_1_mark <> 1`("")
    andalso
OSPF_model'NBR_DB_1_mark ==
1`("10.0.0.22","2-WAY")+
1`("10.0.0.33","2-WAY")

    andalso

OSPF_model'R2_BDR_1_mark <> 1`("")
    andalso
OSPF_model'R2_DR_1_mark <> 1`("")
    andalso
OSPF_model'R2_NBR_DB_1_mark ==
1`("10.0.0.11","2-WAY")+
1`("10.0.0.33","2-WAY")

    andalso

OSPF_model'R3_BDR_1_mark <> 1`("")
    andalso
OSPF_model'R3_DR_1_mark <> 1`("")
    andalso
OSPF_model'R3_NBR_DB_1_mark ==
1`("10.0.0.22","2-WAY")+
1`("10.0.0.33","2-WAY")
    
```



Obr. č.21: Chybové hlásenia v CPN Tools.

Prvotným overením funkčnosti siete je automatické overovanie syntaxe kompilátorom v CPN Tools (znázornené na obrázku *Obr. č.21*). Výhodou tohto kompilátora je, že už pri procese samotnej tvorby priebežne kontroluje funkčnosť modelu pomocou farebného rozlišovania jeho správnych a chybných častí, a zároveň podčiarkuje syntax jazyka ML, ktorý CPN Tools využívajú. Pri zastavení kurzora myši nad zvýraznenú chybu kompilátor poskytne popis a pozíciu zistenej chyby.

11.2 State-space analýza

State-space (SS) analýza, nazývaná taktiež aj ako graf dosiahnuteľnosti (v ang. occurrence graphs, reachability graphs alebo reachability trees), je spôsob analýzy modelov systémov pomocou vytvorenia orientovaných grafov všetkých možných dosiahnuteľných značkovaní modelu. Nástroj na vykonanie SS analýzy je integrovaný priamo v prostredí CPN Tools. Výhodou teda je, že v ktoromkoľvek okamihu je možné prechádzať a sledovať simulátor a SS nástroj. Taktiež ak je nájdený napríklad v strome dosiahnuteľnosti požadovaný stav, je možné si ho v danom okamihu prehliadať aj v simulátore. Nástroj SS má v sebe zabudovanú väčšinu štandardných funkcií napríklad na zisťovanie vlastností Petriho sietí, akými sú dosiahnuteľnosť značkovaní, ohraničenosť, živosť, férovosť a pod. Samozrejme je taktiež možné definovať si vlastné funkcie schopné overovať iné parametre modelu.

V našom prípade sme pre jednoduchosť a zachovanie prehľadnosti modelu vykonali SS analýzu v zapojení dvoch smerovačov v sieti. SS analýza bola vykonaná so základnými parametrami. Kvôli nedostatku dostupných materiálov sa nám nepodarilo zdefinovať podmienku zastavenia SS analýzy (definovania koncového značkovania). Analýzu sme teda vykonali po dobu 120 sekúnd, počas ktorej podľa predošlých pokusov bola vysoká pravdepodobnosť dosiahnutia koncového skonvergovaného stavu. Výsledné štatistiky sú uvedené v nasledujúcich riadkoch:

```
CPN Tools state space report for:  
OSPF_model__FINAL_03_no_hierarchy_w_monitors.cpn
```

```
Statistics
```

```
State Space
```

```
Nodes: 12313  
Arcs: 716260  
Secs: 120  
Status: Partial
```

```
Scc Graph
```

```
Nodes: 7569  
Arcs: 91822
```

Secs: 80

Boundedness Properties

Best Integer Bounds

	Upper	Lower
OSPF_model'BDR 1	1	1
OSPF_model'DR 1	1	1
OSPF_model'NBR_DB 1	1	0
OSPF_model'R2_BDR 1	1	1
OSPF_model'R2_DR 1	1	1
OSPF_model'R2_NBR_DB 1	1	0
OSPF_model'R2_bus 1	4	3
OSPF_model'R2_int_error 1	1	1
OSPF_model'R2_num_of_nbrs 1	1	1
OSPF_model'R2_tmp_nbrs 1	1	1
OSPF_model'Router_bus 1	4	3
OSPF_model'int_error 1	1	1
OSPF_model'loss 1	1	1
OSPF_model'num_of_nbrs 1	1	1
OSPF_model'p1 1	9	0
OSPF_model'p2 1	6	0
OSPF_model'p3 1	5	0
OSPF_model'tmp_nbrs 1	1	1

Best Upper Multi-set Bounds

```

OSPF_model'BDR 1 1`"10.0.0.11"
OSPF_model'DR 1 1`"10.0.0.11"++
OSPF_model'NBR_DB 1 1`("10.0.0.22","2-WAY")++
OSPF_model'R2_BDR 1 1`"10.0.0.11"
OSPF_model'R2_DR 1 1`"10.0.0.22"
OSPF_model'R2_NBR_DB 1 1`("10.0.0.11","2-WAY")++
OSPF_model'R2_bus 1 1`hello_pkt(("10.0.0.22",""))++
hello_pkt(("10.0.0.22","10.0.0.11, "))++
nbr_db(("10.0.0.11","2-WAY"))++
nbr_db(("10.0.0.11","INIT"))++
r_state("READY")++
r_state("new_nbr")++
r_state("old_nbr")++
int_state("DOWN")++
int_state("UP")
OSPF_model'R2_int_error 1
1`95
OSPF_model'R2_num_of_nbrs 1
1`0++

```



```

1`1
    OSPF_model'R2_tmp_nbrs 1
                                1`0++
1`1
    OSPF_model'Router_bus 1
                                1`hello_pkt(("10.0.0.11",""))++
1`hello_pkt(("10.0.0.11","10.0.0.22, "))++
1`nbr_db(("10.0.0.22","2-WAY"))++
1`nbr_db(("10.0.0.22","INIT"))++
1`r_state("READY")++
1`r_state("new_nbr")++
1`r_state("old_nbr")++
1`int_state("DOWN")++
1`int_state("UP")
    OSPF_model'int_error 1
                                1`95
    OSPF_model'loss 1 1`9
    OSPF_model'num_of_nbrs 1
                                1`0++
1`1
    OSPF_model'p1 1 5`hello_pkt(("10.0.0.11",""))++
9`hello_pkt(("10.0.0.11","10.0.0.22, "))++
5`hello_pkt(("10.0.0.22",""))++
9`hello_pkt(("10.0.0.22","10.0.0.11, "))
    OSPF_model'p2 1 2`hello_pkt(("10.0.0.11",""))++
5`hello_pkt(("10.0.0.11","10.0.0.22, "))++
2`hello_pkt(("10.0.0.22",""))++
5`hello_pkt(("10.0.0.22","10.0.0.11, "))
    OSPF_model'p3 1 2`hello_pkt(("10.0.0.11",""))++
5`hello_pkt(("10.0.0.11","10.0.0.22, "))++
2`hello_pkt(("10.0.0.22",""))++
4`hello_pkt(("10.0.0.22","10.0.0.11, "))
    OSPF_model'tmp_nbrs 1
                                1`0++
1`1

Best Lower Multi-set Bounds
    OSPF_model'BDR 1 empty
    OSPF_model'DR 1 empty
    OSPF_model'NBR_DB 1 empty
    OSPF_model'R2_BDR 1 empty
    OSPF_model'R2_DR 1 empty
    OSPF_model'R2_NBR_DB 1
                                empty
    OSPF_model'R2_bus 1 empty
    OSPF_model'R2_int_error 1
                                1`95
    OSPF_model'R2_num_of_nbrs 1
                                empty
    OSPF_model'R2_tmp_nbrs 1
                                empty
    OSPF_model'Router_bus 1
                                empty
    OSPF_model'int_error 1
                                1`95
    OSPF_model'loss 1 1`9
    OSPF_model'num_of_nbrs 1
                                empty

```

```
OSPF_model'p1 1      empty
OSPF_model'p2 1      empty
OSPF_model'p3 1      empty
OSPF_model'tmp_nbrs 1
                    empty
```

Home Properties

Home Markings
None

Liveness Properties

Dead Markings
5867 [9999, 9998, 9997, 9996, 9995, ...]

Dead Transition Instances
None

Live Transition Instances
None

Fairness Properties

Impartial Transition Instances
None

Fair Transition Instances
None

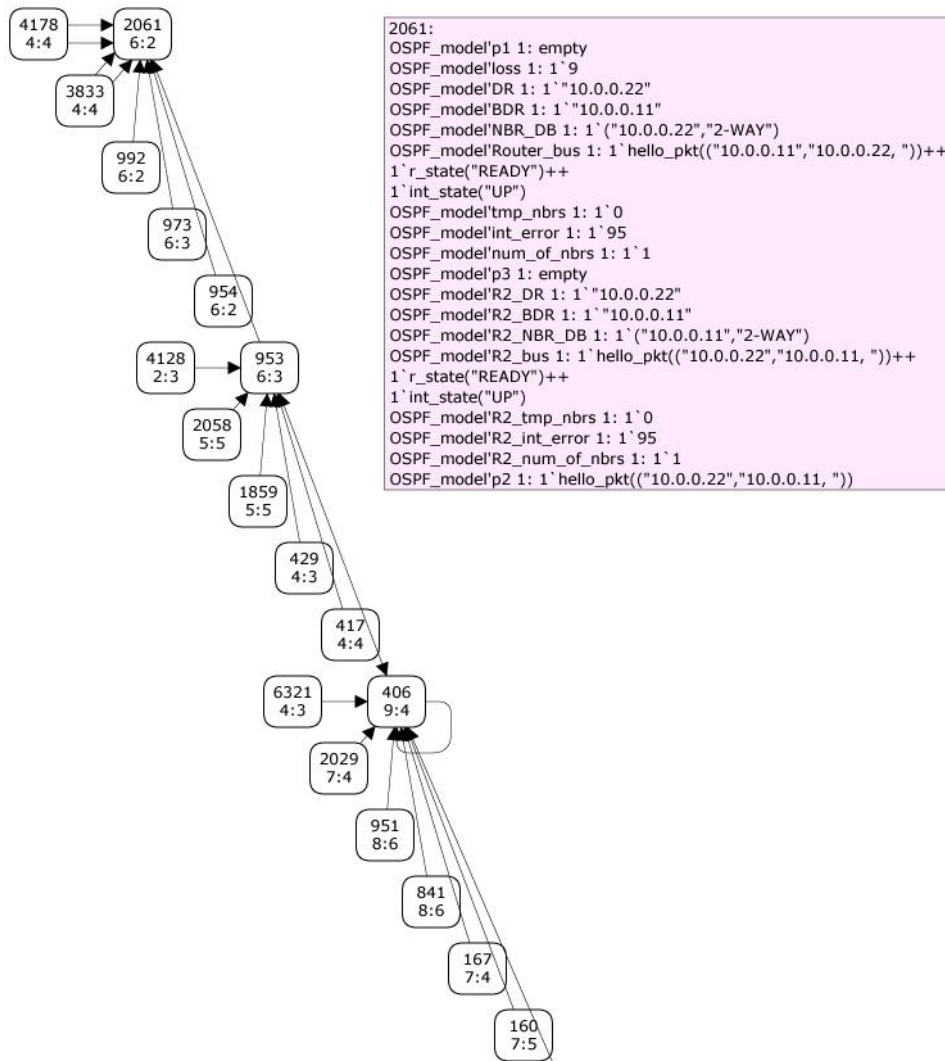
Just Transition Instances
None

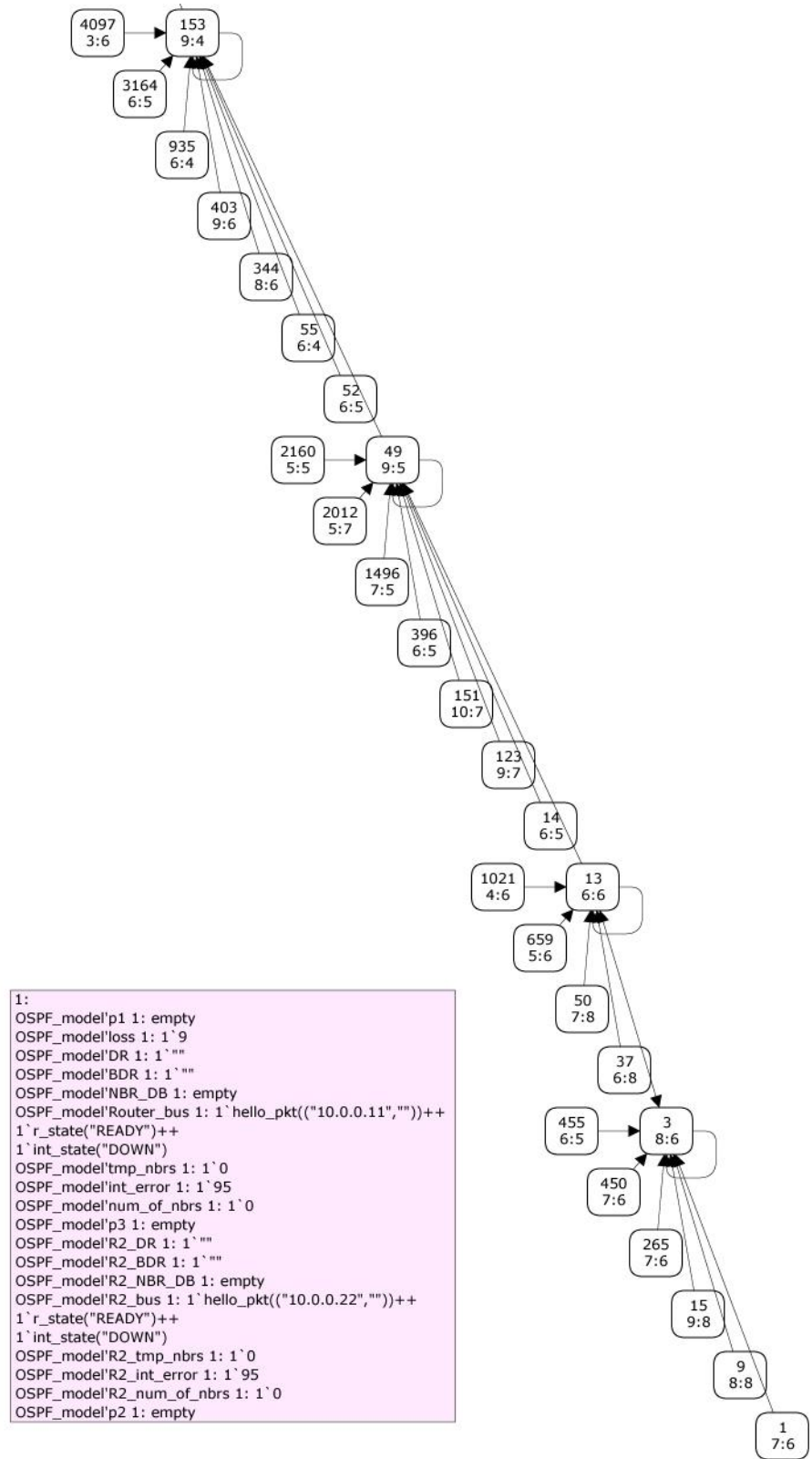
Transition Instances with No Fairness

```
OSPF_model'R2_add_new_nbr 1
OSPF_model'R2_clear_DR 1
OSPF_model'R2_clear_nbr_db 1
OSPF_model'R2_elect_DR 1
OSPF_model'R2_int_DOWN 1
OSPF_model'R2_int_UP 1
OSPF_model'R2_receive_hello 1
OSPF_model'R2_send_hello 1
OSPF_model'R2_treat_old_nbr 1
OSPF_model'add_new_nbr 1
OSPF_model'clear_DR 1
OSPF_model'clear_nbr_db 1
OSPF_model'elect_DR 1
OSPF_model'interface_DOWN 1
OSPF_model'interface_UP 1
OSPF_model'receive_hello 1
OSPF_model'send_hello 1
OSPF_model'transmit 1
OSPF_model'treat_old_nbr 1
```

Graf dosiahnuteľného značkovania je pri takomto komplexnom modeli veľmi zložitý a náročný na orientáciu. Rozhodli sme sa v ňom teda vystopovať iba vetvu, ktorá popisuje najkratšiu cestu od počiatočného stavu k požadovanému koncovému stavu. Graf dosiahnuteľnosti sme zostrojili takým spôsobom, že sme nechali plynúť analýzu všetkých možných stavov po dobu 120 sekúnd a pomocou simulačných nástrojov si určili požadovaný koncový stav siete. Pomocou SS nástroja sme si následne vystopovali daný koncový stav v množine všetkých stavov v grafe dosiahnuteľnosti a zistili si jeho označenie (v našom prípade mal označenie 2061, pozri *Obr. č.22*). Ako vidno v popise daného uzla stromu, ide o naše požadované koncové značkovanie skonvergovanej siete. Z tohto označenia sme potom spätnou analýzou dospeli k počiatočnému značkovaniu s označením 1, v ktorého popise je možné vidieť počiatočné značkovanie nášho modelu siete.

Graf dosiahnuteľnosti koncového značkovania je znázornený na nasledujúcom obrázku:





Obr. č.23: Graf dosiahnuteľného značkovania.

12 Zhodnotenie a budúca práca

V projekte z predmetov Tímový projekt I. a II. sme mali za úlohu analyzovať, implementovať a overiť model nami zvoleného diskretného udalostného systému. Počas hľadania a analýzy konkrétnej aplikácie reálneho systému sme sa orientovali najmä na oblasť informatiky, počítačových systémov a sietí, architektúru softvéru, hardvéru, prípadne automatizačné mechanizmy. Ako vhodný objekt na modelovanie sme si vybrali sieťový smerovací protokol OSPF, ktorý nám poskytoval adekvátnu, priam až nadbytočnú komplexnosť na tvorbu modelu vo vyššom modelovacom prostredí. Ako opisný nástroj sme si zvolili prostredie CPN Tools, ktoré sa zameriava na tvorbu modelov pomocou farebných Petriho sietí poskytujúcich dostatočné možnosti a funkcionality. Voľba tohto nástroja bola pre nás výzvou, pretože disponoval úplne novým prostredím a syntaxou jazyka, s ktorými bolo spočiatku obtiažne sa oboznámiť. Bolo to tak najmä z dôvodu, že oficiálna dokumentácia nástroja CPN Tools nepopisuje detailne všetky funkcie obsiahnuté v jazyku ML, ktorý sa pri modelovaní a simulácií používa. Pre jednoduchšie modely od toho nášho sú však dostupné materiály postačujúce.

S novo naštudovanými vedomosťami sme teda navrhli základný prototyp modelu, ktorý však po dodatočnom analyzovaní nespĺňal podmienky pre korektné fungovanie modelu opisovaného protokolu. Počas predmetu Tímový projekt II. sme sa dostali na úroveň ktorá nám umožnila vytvorenie nového návrhu a poskytla podklady pre implementáciu modelu. Pri návrhu sme sa snažili čím viac sa pridržiavať štandardu opisovaného protokolu. Po dôkladnejšom analyzovaní štandardu sme zistili, že komplexnosť tohto protokolu je vysoko nad rámec rozsahu predmetu Tímový projekt, a preto sme sa rozhodli implementovať iba základnú funkcionality OSPF protokolu, a to nadviazanie komunikácie a vytvorenie susedstva v broadcastovej alebo NBMA sieti. Riešenie sme sa snažili udržať čo najmodulárnejšie a zachovať flexibilitu dopĺňania ľubovoľného počtu smerovačov do siete. Za veľký prínos považujeme osvojenie si poznatkov potrebných na kompletný workflow pri práci s farebnými Petriho sieťami a nástrojom CPN Tools, keďže viacerí členovia tímu sa s témou z podobnej oblasti stretnú aj na predmete Diplomový projekt.

Náš implementovaný model je možné ešte vo veľkej miere rozširovať. Okrem implementovanej funkcie nadviazania spojenia by bolo vhodné dokončiť celkovú funkcionality modelu OSPF protokolu, a to výmenu kompletnej databázy susedov na požiadanie, výmenu všetkých typov LSA paketov, tvorbu smerovacej tabuľky, a i. Najmä z časového hľadiska a čiastočne aj kvôli nedostatku materiálov a používateľských dokumentácií nástroja CPN Tools sme nedokončili optimalizáciu modelu pre state space analýzu a jej prislúchajúce štatistické hodnoty, ako napríklad časovú zložku užitočnú pri porovnávaní výkonu nášho modelu s výkonom modelov iných smerovacích protokolov. Aj napriek týmto nedostatkom veríme, že

naša práca bude vhodným podkladom nielen pre nasledujúce generácie ako zdroj informácií a inšpiráciou pri tvorbe podobných systémov v nástroji CPN Tools.

13 Použitá literatúra

- [1] ADAMUŠČÍNOVÁ, I.: *Koncepty modelov distribuovaných systémov. Diplomová práca. Košice: FEI TU v Košiciach, 2007*
- [2] RADA, T.: *Diagnostika porúch založená na modeloch (automatoch). Diplomová práca, FIIT STU, 2009*
- [3] DORČÁK, L.: *Matematické modelovanie procesov a systémov.*
- [4] Zelenka J., Matejka T.: *Aplikácia teórie diskretných udalostných systémov na analýze a modelovanie reálneho výrobného system, 2010*
- [5] VICEN J., *Typové príklady a využitie Petriho sietí: Bakalárska práca, vedúci BP – Ing. Tomalová, <http://moodle.fiit.stuba.sk/moodle/course/view.php?id=55>*
- [6] Žáry, M.: *Prehľad techník lokalizácie porúch v počítačových sieťach, 2010*
- [7] *Informace pro studenty předmětů PES pro léto 2009/2010.*
<http://147.229.9.23/study/courses/PES/public/index.html.en#Aktuality>
- [8] Hollý, J., *STOCHASTICKÉ PETRIHO SIETE II, STU Bratislava, 2009.*
- [9] *Petri Nets Tools and software*
<http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/>
- [10] *OSPF Overview, History, Standards and Versions, (5.11.2010)*
http://www.tcpipguide.com/free/t_OSPFOverviewHistoryStandardsandVersions.htm
- [11] *Open Shortest Path First, (5.11.2010)*
<http://www.cs.vsb.cz/grygarek/SPS/lect/OSPF/ospf.html>
- [12] *OSPF General Operation and Message Types, (5.11.2010)*
http://www.tcpipguide.com/free/t_OSPFGeneralOperationandMessageTypes.htm
- [13] *CCIE Talk: OSPF LSA Types, (6.11.2010)*
<http://www.ccietalk.com/2008/07/13/ospf-lsa-types>
- [14] *Cisco IP Routing – OSPF Neighbor States, (5.11.2010)*
http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094050.shtm

14Prílohy

14.1 Dokumentácia riadenia



Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 812 19 Bratislava



Diagnostika porúch diskretných udalostných systémov založená na
modeloch a aplikácie v informatike

Dokumentácia riadenia

(Tím č.5)

Bc. Tomáš Bartek
Bc. Jozef Ferenčík
Bc. Lukáš Humený
Bc. Peter Kubanda
Bc. Pavol Kubík
Bc. Michal Kurtý
Bc. Michal Zapatický

Vedúci projektu: Ing. Jana Flochová

Študijný program: Počítačové a komunikačné systémy a siete

Štúdium: inžinierske

Ročník: 1.

Akademický rok: 2010/2011

Predmet: Tímový projekt I,II

OBSAH

História dokumentu	Chyba! Záložka není definována.
Úvod	Chyba! Záložka není definována.
1 Ponuka	Chyba! Záložka není definována.
1.1 Zadanie	Chyba! Záložka není definována.
1.2 Motivácia	Chyba! Záložka není definována.
1.3 Analýza zadania a hrubý návrh riešenia	Chyba! Záložka není definována.
1.4 Členovia tímu	Chyba! Záložka není definována.
Bc. Tomáš Bartek	Chyba! Záložka není definována.
Bc. Jozef Ferenčík	Chyba! Záložka není definována.
Bc. Lukáš Humený	Chyba! Záložka není definována.
Bc. Peter Kubanda	Chyba! Záložka není definována.
Bc. Pavol Kubík	Chyba! Záložka není definována.
Bc. Michal Kurtý	Chyba! Záložka není definována.
Bc. Michal Zapatický	Chyba! Záložka není definována.
1.5 Témy podľa priority záujmu	Chyba! Záložka není definována.
1.6 Stretnutia tímu v ZS	Chyba! Záložka není definována.
1.7 Stretnutia tímu v LS	Chyba! Záložka není definována.
2.Plán projektu	Chyba! Záložka není definována.
2.1 Plán na zimný semester	Chyba! Záložka není definována.
2.2 Plán na letný semester	Chyba! Záložka není definována.
3 Úlohy v tíme	Chyba! Záložka není definována.
3.1 Komunikácia v tíme	Chyba! Záložka není definována.
3.2 Stretnutie členov tímu mimo pravidelných strenutí	Chyba! Záložka není definována.
3.3 Mobilný telefón (Skype)	Chyba! Záložka není definována.
3.4 Chat (ICQ)	Chyba! Záložka není definována.
3.5 Google groups	Chyba! Záložka není definována.
3.6 E-mail	Chyba! Záložka není definována.
3.7 Funkcie členov tímu	Chyba! Záložka není definována.
3.8 Dlhodobé úlohy	Chyba! Záložka není definována.
3.9 Krátkodobé úlohy	Chyba! Záložka není definována.
4 Záznamy o stretnutiach	Chyba! Záložka není definována.
4.1 Zápisnica 01	Chyba! Záložka není definována.
4.2 Zápisnica 02	Chyba! Záložka není definována.
4.3 Zápisnica 03	Chyba! Záložka není definována.
4.4 Zápisnica 04	Chyba! Záložka není definována.
4.5 Zápisnica 05	Chyba! Záložka není definována.
4.6 Zápisnica 06	Chyba! Záložka není definována.
4.7 Zápisnica 07	Chyba! Záložka není definována.
4.8 Zápisnica 08	Chyba! Záložka není definována.
4.9 Zápisnica 09	Chyba! Záložka není definována.
4.10 Zápisnica 10	Chyba! Záložka není definována.
4.11 Zápisnica 11	Chyba! Záložka není definována.
4.12 Zápisnica 12	Chyba! Záložka není definována.
4.13 Zápisnica 13	Chyba! Záložka není definována.

4.14 Zápiska 14	Chyba! Záložka není definována.
4.15 Zápiska 15	Chyba! Záložka není definována.
4.16 Zápiska 16	Chyba! Záložka není definována.
4.17 Zápiska 17	Chyba! Záložka není definována.
4.18 Zápiska 18	Chyba! Záložka není definována.
4.19 Zápiska 19	Chyba! Záložka není definována.
4.20 Zápiska 20	Chyba! Záložka není definována.

HISTÓRIA DOKUMENTU

Dátum zmeny	Verzia	Opis zmeny
11.10.2010	1.0	Vytvorenie základnej formy dokumentu, vytvorenie názvov jednotlivých kapitol, pridanie ponuky
11.10.2010	1.1	Pridaná zápisnica01 a pridaný plán projektu
18.10.2010	1.2	Pridaná zápisnica02
25.10.2010	1.3	Pridaná zápisnica03 a pridané rozdelenie úloh členom tímu
05.11.2010	1.4	Pridaná zápisnica04
08.11.2010	1.5	Pridaná zápisnica05
15.11.2010	1.6	Pridaná zápisnica06
22.11.2010	1.7	Pridaná zápisnica07
29.11.2010	1.8	Pridaná zápisnica08
09.12.2010	1.9	Pridaná zápisnica09
26.02.2011	2.0	Pridaná zápisnica10
03.03.2011	2.1	Pridaná zápisnica11
10.03.2011	2.2	Pridaná zápisnica12
18.03.2011	2.3	Pridaná zápisnica13
23.03.2011	2.4	Pridaná zápisnica14
30.03.2011	2.5	Pridaná zápisnica15
06.04.2011	2.6	Pridaná zápisnica16
13.04.2011	2.7	Pridaná zápisnica17
20.04.2011	2.8	Pridaná zápisnica18
27.04.2011	2.9	Pridaná zápisnica19
03.05.2011	3.0	Pridaná zápisnica20

Tabuľka 1 História dokumentu

ÚVOD

Dokumentácia riadenia projektu zahŕňa všetky informácie a materiály, ktoré vznikajú počas riešenia projektu. Dokument obsahuje presné znenie ponuky, ktorú sme vypracovali na začiatku zimného semestra, keď sme sa uchádzali o tento projekt. Ďalej dokumentácia zahŕňa rozpis jednotlivých členov tímu a ich funkcie pri tvorbe projektu, plán projektu na zimný i letný semester, a samozrejme aj dlhodobé a krátkodobé úlohy pridelené jednotlivým členom tímu. V tomto dokumente sú uvedené aj všetky zápisnice zo spoločných stretnutí tímu s vedúcim projektu.

1 PONUKA

Ponuka bola vytvorená pre dôvod udelenia témy “Programová podpora pre sieťový simulátor“ pre náš tím č.5. Našou snahou bolo presvedčiť za pomoci tejto ponuky vedúceho predmetu, že nami vybraná téma by mala byť riešená práve našim tímom. V nasledujúcich podkapitolách je preto uvádzaná ponuka, motivácia, analýza ako aj hrubý návrh riešenia pre túto pre nás prioritnú tému. Nakoniec nám bola pridelený projekt s názvom “Diagnostika porúch diskretných udalostných systémov založená na modeloch a aplikácie v informatike“

1.1 ZADANIE

Programová podpora pre sieťový simulátor

Analyzujte možnosti súčasných sieťových simulátorov pričom sa zamerajte na popredné simulátory ns2 a ns3.

Analýzu upriamte na simulačné možnosti vybraných typov počítačových a komunikačných sietí so zameraním sa na prípravu a vyhodnotenie prebehnutých simulácií. Sústreďte sa na typické požiadavky vyhodnocovania simulácií vybraných typov sietí.

Na základe vykonanej analýzy navrhnete a následne implementujete systém, ktorý používateľovi umožní ľahšiu prípravu a vyhodnotenie prebehnutých simulácií. Systém musí poskytovať čo najuniverzálnejšie prostredie pre vyhodnocovanie a zobrazovanie priebehov a štatistík typických parametrov sledovaných pri simuláciách vybraných typov sietí.

1.2 MOTIVÁCIA

Každý člen tímu sa počas štúdia orientuje hlavne na oblasť komunikačných sietí, o čom svedčí aj fakt, že väčšina členov má absolvované semestre kurzu CISCO CCNA. Nakoľko všetci členovia tímu študujeme študijný odbor Počítačové a komunikačné systémy a siete a teda v danej oblasti máme dostatočné vedomosti a skúsenosti, rozhodli sme vypracovať ponuku na tému „Programová podpora pre sieťový simulátor“. Naš výber preto jednohlasne padol na túto tému, v ktorej môžeme v čo najväčšej miere využiť naše vedomosti z bakalárskeho štúdia.

Významným motivačným faktorom je aj fakt, že jeden z členov vypracoval a úspešne obhájil bakalársku prácu na tému „Výkonnosť protokolu TCP v bezdrôtových senzorových sieťach“, vďaka ktorej máme výhodnú východiskovú pozíciu pri riešení tohto projektu.

Vyriešenie projektu je postavené na spoločnom zapojení síl, vzájomnej spolupráci a zodpovednosti všetkých členov tímu pri plnení stanovených úloh. Práve táto vzájomná podpora je hlavnou motivačnou silou, ktorá bude viesť náš tím k

úspešnému návrhu a implementácii systému, ktorý budúcim používateľom umožní ľahšiu prípravu a vyhodnotenie prebehnutých simulácií, a teda takto vytvorený systém bude mať aj reálne využitie. Veríme, že kombináciou našich nápadov a skúseností dokážeme vytvoriť projekt, ktorý bude spĺňať všetky stanovené požiadavky a stane sa tak prínosom pre danú oblasť.

1.3 ANALÝZA ZADANIA A HRUBÝ NÁVRH RIEŠENIA

Simulátor NS2 resp. NS3 je momentálne jeden z najuniverzálnejších voľne dostupných simulátorov na simulovanie sieťových modelov. Jedná sa o diskretný udalostný systém, pomocou ktorého je možné navrhnuť rôzne typy topológií. Pracuje na princípe prepojenia koncových uzlov (modely sieťových prvkov), ktoré sú navzájom prepojené rôznym typom liniek. Medzi uzlami prebieha cez prenosové jednotky (rámce, segmenty..) dopredu navrhnutá dátová komunikácia.

Simulačné možnosti NS zahrňujú bežné káblové, bezdrôtové alebo aj satelitné siete. Podporované sú viaceré protokoly zásobníka TCP/IP. Návrh siete je vytváraný pomocou programovacieho jazyka Tcl. Výstupy z vykonanej simulácie sa ukladajú do textového súboru ako prenosové udalosti vykonávané v časovej postupnosti. Najčastejšie sa jedná o časový popis pohybu rámcov medzi uzlami. Tieto logy sa analyzujú a spracúvajú pomocou skriptovacích jazykov príp. iných filtračných nástrojov. Spracované výstupy sa následne zobrazujú pomocou grafov, štatistík prípadne porovnávacích tabuliek.

V počítačových sieťach sa najčastejšie vyhodnocujú parametre ako je množstvo prenesených dát, priepustnosť (počet bajtov za sek), stratovosť (počet zahodených rámcov) prípadne oneskorenie prenosu (milisekundy) na danom uzle alebo rozhraní. Na základe nich môžeme porovnávať spoľahlivosť, rýchlosť a stabilitu siete.

Nad týmto spracovávaním údajov a ich vyhodnocovaním bude potrebné vytvoriť prehľadné užívateľské prostredie, cez ktoré bude možné jednoducho vytvárať sieťovú topológiu s požadovanými parametrami, simulovať jej správanie pri zadaných podmienkach (obmedzenia, rozšírenia) a výsledky jednoducho graficky vyhodnotiť alebo tabuľkovo porovnať jednotlivé údaje.

1.4 ČLENOVIA TÍMU

Nasledujúca kapitola predkladá základné informácie o tíme a jeho členoch. Tím pozostáva zo šiestich študentov zaujímajúcich sa o rôzne oblasti informačných technológií. Okrem vedomostí a zručností nadobudnutých počas absolvovaného bakalárskeho štúdia na Fakulte informatiky a informačných technológií sú v charakteristikách uvedené aj vedomosti nadobudnuté popri štúdiu týkajúce sa tvorby komplexnejších projektov, prípadne skúsenosti získané v profesionálnych pracovných podmienkach.

BC. TOMÁŠ BARTEK



Pozícia v tíme: tvorba modelov a simulácia
Absolvent bakalárskeho štúdia na FIIT STU, odbor PSS
Absolvované predmety súvisiace s vybraným projektom:
Počítačové siete 1, Počítačové siete 2,
Absolvované kurzy a samo štúdium: CCNA, CCNP
Zapísané predmety: Bezdrôtové komunikačné systémy
Znalosť programovacích jazykov C, Java, Adobe
Actionscript (Adobe Flash CS4)
Bakalársky projekt na tému: Multimediálny programový
systém na podporu výučby Petriho sietí
Vo voľnom čase sa venuje návrhu a správe sietí a
programovaniu v jazyku Actionscript.
email: tomas.bartek@ynet.sk

BC. JOZEF FERENČÍK



pozícia v tíme: návrh modelov a automatov
znalosť programovacích jazykov C, Java, PHP, CSS
tvorba webových stránok, návrh grafiky
absolvoval predmety súvisiace s vybraným projektom:
Počítačové siete 1, Počítačové siete 2
bakalársky projekt na tému: Grafická vizualizácia funkčného
generovania testov logických obvodov
e-mail: jferencik@gmail.com

BC. LUKÁŠ HUMENÝ



pozícia v tíme: vyhľadávanie a overovanie chýb v modeloch
programovacie jazyky: C,C++, Tcl
súvisiace absolvované predmety: Operačné systémy,
Počítačové systémy a siete1,2
súvisiace zapísané predmety: Bezdrôtové komunikačné
systémy
bakalárska práca: Výkonnosť protokolu TCP v bezdrôtových
senzorových sieťach – zameranie na simulovanie a analýzu
rôznych typov sieťových topológií v simulátore NS2

iné znalosti: praktické skúsenosti so sieťovým simulátorom NS2, práca s reálnymi sieťovými technológiami a ich konfigurácia (certifikáty CISCO CCNA)
email: lukas.humeny@gmail.com

BC. PETER KUBANDA



pozícia v tíme: vývoj webovej stránky, grafický dizajn
programovacie znalosti: C, C++, C#, PHP, MySQL
absolvované predmety súvisiace s vybraným projektom: Počítačové siete 1, Počítačové siete 2
zapísané predmety: Bezdrôtové komunikačné systémy
bakalárska práca: Vizualizácia BFMA stromu na PC
vo voľnom čase sa venuje tvorbe webových stránok
email: peter.kubanda@gmail.com

BC. PAVOL KUBÍK



pozícia v tíme: vývoj webovej stránky, grafický dizajn
programovacie znalosti: procedurálne programovanie, objektovo-orientované programovanie, PHP+MySQL
absolvované predmety súvisiace s vybraným projektom: Počítačové siete 1, Počítačové siete 2
zapísané predmety: Bezdrôtové komunikačné systémy
bakalárska práca: Porovnanie virtualizačných nástrojov
email: lejko4@gmail.com

BC. MICHAL KURTÝ



pozícia v tíme: vedúci tímu, koordinácia a riadenie tímu
programovacie jazyky: C, C++, PHP, MySQL
má zručnosti v správe sieťových zariadení, štyri semestre CCNA
vo voľnom čase pracuje s 3D a 2D grafikou
téma bakalárskej práce: Oneskorenie prenosu dát v TCP/IP sieťach
email: kurty.miso@gmail.com

BC. MICHAL ZAPATICKÝ



pozícia v tíme: vyhľadávanie a overovanie chýb v modeloch programovacie jazyky: procedurálne programovanie, VHDL
 absolvované predmety súvisiace s vybraným projektom: Počítačové siete 1, Počítačové siete 2, Bezdrôtové komunikačné systémy, Diagnostika a spoľahlivosť digitálnych systémov
 bakalárska práca: Syntéza mostíkových kontaktovo-diódových dvojpólov
 email: zapaticky.m@gmail.com

1.5 TÉMY PODĽA PRIORITY ZÁUJMU

1. Programová podpora pre sieťový simulátor - *Ing. P. Magula*
2. Diagnostika porúch diskretných udalostných systémov založená na modeloch a aplikácie v informatike - *Ing. J.Flochová, PhD.*
3. Vývoj vzorových aplikácií pre IPTV v prostredí Microsoft Mediaroom Framework - *Doc. Ing. I. Kotuliak, PhD.*

1.6 STRETNUTIA TÍMU V ZS

	7.00 7.50	8.00 8.50	9.00 9.50	10.00 10.50	11.00 11.50	12.00 12.50	13.00 13.50	14.00 14.50	15.00 15.50	16.00 16.50	17.00 17.50	18.00 18.50	19.00 19.50	20.00 20.50
Po									STRETNUTIE TÍMU					
Ut														
St														
Št														
Pi														

1.7 STRETNUTIA TÍMU V LS

	7:00 7:50	8:00 8:50	9:00 9:50	10:00 10:50	11:00 11:50	12:00 12:50	13:00 13:50	14:00 14:50	15:00 15:50	16:00 16:50	17:00 17:50	18:00 18:50	19:00 19:50	20:00 20:50
Po			x	x		x	x			x	x	x	x	
Ut									x	x	x	x	x	
St			x	x	x	Stretnutie Tímu		x	x	x	x	x	x	
Št	x	x	x	x	x	x	x	x	x	x	x	x		
Pi		x	X											

2.PLÁN PROJEKTU

V tejto kapitole je uvedený časový plán projektu. Plán prác je rozdelený na jednotlivé týždne v semestri.

2.1 PLÁN NA ZIMNÝ SEMESTER

1. týždeň

- Zostavenie tímu
- Oboznámenie sa so zadanými projektmi
- Výber témy a vytvorenie ponuky k vybranej téme

2. týždeň

- Prezentácia ponuky

3. týždeň

- Pridelenie 2.prioritnej témy
- Prvé stretnutie s pedagogickým vedúcim

4. týždeň

- Rozdelenie úloh jednotlivým členom tímu
- Dohodnutie konzultačných hodín projektu na pondelok 15:00 v softvérovom štúdiu
- Diskusia o téme a úlohách projektu, návrh konkrétnych tém na vypracovanie

5. týždeň

- Prednáška o modelovaní systémov pomocou Petriho sietí
- Vytvorenie prvotnej verzie webovej prezentácie projektu

6. týždeň

- Pokračovanie prednášky o modelovaní systémov pomocou Petriho sietí
- Aktualizácia webovej prezentácie projektu

7. týždeň

- Stretnutie členov tímu
- Modelovacie nástroje
- Konzultácia v softvérovom štúdiu
- Aktualizácia webovej prezentácie projektu

8. týždeň

- Odovzdanie dokumentácie analýzy problému a špecifikácie požiadaviek riešenia spolu s hrubým návrhom
- Prerozdelenie úloh medzi jednotlivých členov na vypracovanie teórie dokumentu
- Aktualizácia webovej prezentácie projektu

9. týždeň

- Kontrola dokumentu

- Návrh ďalšieho postupu
- Hodnotenie druhého tímu
- Aktualizácia webovej prezentácie projektu

10. týždeň

- Návrh hrubého modelu
- Zoznámenie sa s nástrojmi PIPE2 a CPN Tools
- Aktualizácia webovej prezentácie projektu

11. týždeň

- Vytvorený hrubý model sa pokúsime vytvoriť v simulátore CPN Tools
- Riešenie problematiky OSPF a jeho využitie ako modelu v Petriho sieťach
- Aktualizácia webovej prezentácie projektu

12. týždeň

- Namodelovanie a doplnenie vytvoreného prototypu modelu OSPF o funkcionálnosť
- Dokončenie dokumentácie za 2. kvartál a dokumentácie riadenia
- Prezentovanie prototypu oponentskému tímu
- Aktualizácia webovej prezentácie projektu

2.2 PLÁN NA LETNÝ SEMESTER

1. týždeň

- Zhodnotenie činnosti za minulý semester
- Aktualizácia webovej prezentácie projektu

2. týždeň

- Rozširovanie prototypu sieťového protokolu v CPN nástroji
- Aktualizácia webovej prezentácie projektu

3. týždeň

- Rozširovanie prototypu sieťového protokolu v CPN nástroji
- Aktualizácia webovej prezentácie projektu

4. týždeň

- Implementácia podmienok a polí v modeli v nástroji CPN Tools
- Aktualizácia webovej prezentácie projektu

5. týždeň

- Vytvorenie hierarchického modelu v nástroji CPN Tools
- Aktualizácia webovej prezentácie projektu

6. týždeň

- Rozširovanie hierarchického modelu v nástroji CPN Tools
- Aktualizácia webovej prezentácie projektu

7. týždeň

- Implementácia DR a BDR informácie do Hello packetov
- Aktualizácia webovej prezentácie projektu

8.týždeň

- Implementácia štatistiky (stratovosť paketov, zlyhanie smerovača) do modelu
- Aktualizácia webovej prezentácie projektu

9.týždeň

- Dokončenie kompletnej hierarchie modelovaného OSPF
- Aktualizácia webovej prezentácie projektu

10.týždeň

- Dokončenie a vylepšenie hierarchického modelu v nástroji CPN Tools
- Dokončenie štatistík (stratovosť paketov, zlyhanie smerovača) v modeli
- Aktualizácia webovej prezentácie projektu

11.týždeň

- Dokončenie a posledné úpravy no modeli OSPF v CPN Tools
- Dokončenie všetkých častí dokumentácie
- Aktualizácia webovej prezentácie projektu

12.týždeň

- Kontrola funkčnosti celého modelu OSPF v CPN Tools
- Príprava na posudok oponentského tímu a obhajobu projektu
- Aktualizácia webovej prezentácie projektu



3 ÚLOHY V TÍME

Rozdelenie úloh medzi jednotlivých členov v tíme. Úlohy boli rozdelené na krátkodobé a dlhodobé. K úlohám sú pripojené aj jednotlivé funkcie členov tímu.

3.1 KOMUNIKÁCIA V TÍME

Tímová komunikácia je dôležitá pre proces riadenia tímu. Rozdelíme si ju podľa spôsobu akým sa komunikuje a popíšeme si jej jednotlivé vlastnosti, postupy pri komunikácií.

Pravidelné stretnutia sú stretnutia, ktoré sa konajú raz týždenne za účasti všetkých členov tímu a pedagogického vedúceho projektu. Na týchto stretnutiach sa rozoberajú úlohy jednotlivých členov a prezentuje sa ich progres a postup zadaného riešenia. Výstupom každého stretnutia je zápisnica, ktorá slúži na detailný popis priebehu takéhoto stretnutia.

Priebeh stretnutia:

1. oboznámenie všetkých členov so stavom riešení a následná diskusia.
2. predstretie nových návrhov a diskusia okolo nich.
3. zadanie nových, prípadne predĺženie termínu odovzdania starých úloh.

3.2 STRETNUTIE ČLENOV TÍMU MIMO PRAVIDELNÝCH STRETNUTÍ

Nie je povinné, aby sa na nich zúčastňovali všetci členovia. V prípade potreby konzultácie riešenia sa členovia tímu, ktorých sa to týka stretnú a dohodnú sa na riešení úlohy, ktorá im bola zadaná.

3.3 MOBILNÝ TELEFÓN (SKYPE)

Slúži na okamžité upovedomenie členov tímu. Využíva sa hlavne v prípade nutnosti komunikácie alebo odovzdania informácie, ktorá je dôležitá pre potreby tímu.

3.4 CHAT (ICQ)

Slúži na interaktívnu textovú komunikáciu s členmi tímu, ktorí nemohli byť účastní na stretnutí tímu, prípadne na ďalšiu diskusiu k problematike a k riešeniu zadaných úloh pre jednotlivých členov.

3.5 GOOGLE GROUPS

Google stránka bola vytvorená za účelom komunikácie a diskusie k celému projektu ako aj na zdieľanie dokumentov, literatúry a odkazov.

3.6 E-MAIL

Služi na rýchlu textovú komunikáciu a rýchle preposielanie dokumentov.

3.7 FUNKCIE ČLENOV TÍMU

Tomáš Bartek – návrh modelov a simulácia, priebežná dokumentácia

Jozef Ferenčík – návrh modelov a simulácia, priebežná dokumentácia

Lukáš Humený – vyhľadávanie a overovanie chýb v modeloch, tvorba zápisníc

Peter Kubanda – vývoj webovej stránky, grafický dizajn, priebežná dokumentácia

Pavol Kubík – vývoj a aktualizácia webovej stránky, grafický dizajn, dokumentácia riadenia

Michal Kurtý – vedúci tímu, koordinácia a riadenie tímu, kontrola dodržiavania plánu, priebežná dokumentácia

Michal Zapatický – vyhľadávanie a overovanie chýb v modeloch, priebežná dokumentácia

3.8 DLHODOBÉ ÚLOHY

V nasledovnom prehľade je uvedené prerozdelenie dlhodobých úloh v tíme (pozri tab.2).

Typ úlohy	Tomáš Bartek	Jozef Ferenčík	Lukáš Humený	Peter Kubanda	Pavol Kubík	Michal Kurtý	Michal Zapati.
Aktualizácia web stránky					100%		
Dokumentácia riadenia					100%		
Brainstorming (vymýšľanie nových nápadov)	14,29%	14,29%	14,29%	14,29%	14,29%	14,29%	14,29%
Konzultácie, návrhy riešenia	14,29%	14,29%	14,29%	14,29%	14,29%	14,29%	14,29%
Tvorba zápisníc			37%		60%	3%	
Návrh modelov a simulácia	40%		40%			20%	
Vyhľadávanie a overovanie chýb v			20%	20%	20%	20%	20%

modeloch							
Dokumentácia	14,29%	14,29%	14,29%	14,29%	14,29%	14,29%	14,29%

Tabuľka 2: Dlhodobé úlohy

3.9 KRÁTKODOBÉ ÚLOHY

V tejto podkapitole je uvedená účasť jednotlivých členov tímu na konkrétnych úlohách v rámci projektu (pozri tab.3).

Typ úlohy	Tomáš Bartek	Jozef Ferenčík	Lukáš Humený	Peter Kubanda	Pavol Kubík	Michal Kurtý	Michal Zapati.
Tvorba web stránky				50%	50%		
Dizajn web stránky				50%	50%		
Nahratie web stránky na web server					100%		
Pravidlá spoločnej práce na projekte, vymyslieť meno tímu	16,66%	16,66%	16,66%	16,66%	16,66%	16,66%	
Vytvorenie google groups			50%			50%	
Vytvorenie ponuky pre vybranú tému	16,66%	16,66%	16,66%	16,66%	16,66%	16,66%	
Osnova špecifikácie a analýzy prvej časti projektu						100%	
Modelovanie a vlastnosti Petriho sietí	100%						
Základy modelovania a systémov Petriho siete		100%					
Modelovacie nástroje			100%				
Špecifikácia a návrh	100%						

Kontrola a doplnenie zápisníc						100%	
Zoskupenie jednotlivých častí a vytvorenie kompletnej dokumentácie							100%
Posudok dokumentácie tímu č.9-1.kvartál	16,66%	16,66%	16,66%	16,66%	16,66%	16,66%	16,66%
Vytvorenie hrubého modelu OSPF	16,66%	16,66%	16,66%	16,66%	16,66%	16,66%	16,66%
Namodelovanie OSPF v CPN Tools	16,66%	16,66%	16,66%	16,66%	16,66%	16,66%	16,66%
Stavové diagramy modelu					100%		
Popis dátových štruktúr, rozdelenie na moduly, základná funkcionality	100%						
Vytvorenie prototypu modelu v CPN Tools			100%				
Popis budúcej práce, plán na letný semester		100%					
Skompletizovanie dokumentácie riadenia a zápisníc						100%	
Prezentácia prototypu oponentskému tímu							100%
Posudok	16,66%	16,66%	16,66%	16,66%	16,66%	16,66%	16,66%

dokumentácia tímu č.9-2.kvartál		%		
Tvorba jednotlivých verzií modelov	40%	40%	20%	
Finálna dokumentácia			100%	
Napálenie webovej prezentácie na CD			100%	

Tabuľka 3: Krátkodobé funkcie

4 ZÁZNAMY O STRETNUTIACH

Stretnutia tímu prebiehali spolu s našou vedúcou projektu, ktorou je Ing. Jana Flochová. V tejto kapitole sa nachádzajú všetky zápisnice z uskutočnených stretnutí. Zápisnice sú označené ako 0x a zatiaľ všetky sú zo zimného semestra. Úlohy v zápisniciach sú označované spôsobom X,Y,Z (X – dátum, týždeň semestra, Y - stretnutie, Z – program stretnutia).

4.1 ZÁPISNICA 01

Úvodné stretnutie členov tímu (4. týždeň semestra)

Dátum stretnutia : 11.10.2010

Program stretnutia:

1. Časový plán stretávania

- dohodnutie konzultačných hodín projektu na pondelok 15:00 v softvérovom štúdiu

2. Konkretizácia témy projektu

- diskusia o téme a úlohách projektu, návrh konkrétnych tém na vypracovanie

3. Budúce stretnutie

- Ing. Jana Flochová nám na budúcom stretnutí povie nejaké teoretické základy ohľadom modelovania Petriho sietí

Prítomní:

Tomáš Bartek
Jozef Ferenčík
Lukáš Humený
Peter Kubanda
Pavol Kubík
Michal Kurtý

Ing. Jana Flochová, PhD.

Neprítomní:

vypracoval: Lukáš Humený 18.10:2010

4.2 ZÁPISNICA 02

Druhé stretnutie (5. týždeň semestra)

Dátum stretnutia : 18.10.2010

Program stretnutia:

1. Prednáška o modelovaní systémov pomocou Petriho sietí

- teoretické základy Petriho sietí
- Baesovo pravidlo
- závislá pravdepodobnosť
- Markulovský reťazec
- strom dosiahnuteľných pravdepodobností (odhaľovanie chýb)
- stochastické siete (neurčitý čas)
- inhibítori
- queing theory

2. Nahratie webovej stránky

- v priebehu týždňa nahráme našu webovú stránku na určené miesto pre všetky tímové projekty

<http://labss2.fiit.stuba.sk/TeamProject/2010/team05pss/>

3. Budúce stretnutie

- Ing. Jana Flochová dokončí prednášku o teoretických základoch modelovania pomocou Petriho sietí

Prítomní:

Tomáš Bartek
Jozef Ferenčík
Lukáš Humený
Peter Kubanda
Pavol Kubík
Michal Kurtý

Ing. Jana Flochová, PhD.

Neprítomní:

vypracoval: Lukáš Humený 21.10:2010

4.3 ZÁPISNICA 03

Tretie stretnutie (6. týždeň semestra)
Dátum stretnutia : 25.10.2010

Program stretnutia:

1. Pokračovanie prednášky o modelovaní systémov pomocou Petriho sietí

- nepodmienená pravdepodobnosť
- podmienená pravdepodobnosť
- Baesovo pravidlo

2. Nahratie webovej stránky

- stránka tímového projektu nahratá na server s adresou:
<http://labss2.fiit.stuba.sk/TeamProject/2010/team05pss/>
- pridané zápisnice z predošlých stretnutí

3. Budúce stretnutie

- Ing. Jana Flochová dokončí prednášku o metódach aplikácií a priblíži príklady pomocou bipartitných grafov a získa informácie ohľadom posudzovania prác konkurenčných tímov
- špecifikujeme si konkrétne príklady a aplikácie sieťových služieb, ktoré implementujeme v práci a porovnáme s už existujúcimi aplikáciami
- špecifikujeme si obsah výsledného dokumentu a hrubý obsah kapitol

Prítomní:

Tomáš Bartek
Jozef Ferenčík
Peter Kubanda
Pavol Kubík
Michal Kurtý

Ing. Jana Flochová, PhD.

Neprítomní:

Lukáš Humený

vypracoval: Michal Kurtý 27.10.2010

4.4 ZÁPISNICA 04

Štvrté stretnutie (7. týždeň semestra)

Dátum stretnutia : 5.11.2010

Program stretnutia:

1. Stretnutie členov tímu (30 min)

- pred oficiálnou konzultáciou sme mali pol hodinové stretnutie členov tímu, na ktorom sme sa dohodli na finálnej podobe témy.
- vybrali sme si analýzu a simuláciu sieťového smerovacieho protokolu OSPF.
- jeho model aj simuláciu chýb budeme vytvárať pomocou Petriho sietí vo vybranom simulátore.

2. Modelovacie nástroje (30 min)

- Ing. Flochová nám ukázala príklady simulácií časových automatov v prostredí programu Uppaal a ich verifikáciu
- odporučila nám nasledovné programy na modelovanie, keďže sme sa rozhodli využiť Petriho siete
 - ✓ CPN tools (farebné petriho siete)
 - ✓ HPSim (stochastické petriho siete)
 - ✓ PIPE2 (stochastické petriho siete)

3. Konzultácia v softvérovom štúdiu (1h)

- prebrali sme organizačné pokyny ohľadom vedenia konzultácií
- zamerali sme na teóriu Markových reťazcov, výpočty pravdepodobností, použitie inhibítorov
 - ✓ Ing. Flochová nám poskytne súhrn teórie a vzorcov potrebných pri modelovaní

- Michal Kurtý pripravil osnovu špecifikácie a analýzy prvej časti tímového projektu
 - ✓ Ing. Flochová požadovala doplniť ešte kapitoly týkajúce sa toho čo chceme analyzovať a čo chceme modelom zistiť
- Ohľadom modelovania a vlastností Petriho sietí nám Ing. Flochová poskytne knižku v elektronickej podobe od českých kolegov
 - ✓ Tomáš Bartek má k danej téme názornú prednášku

4. Budúce stretnutie

- aktualizácia webu
- doplníme špecifikáciu
- vyberieme si vhodný simulačný nástroj
- prerozdélíme si úlohy medzi členov tímu
- Ing. Flochová povie niečo o Queue teórii

Prítomní:

Tomáš Bartek
Jozef Ferenčík
Lukáš Humený
Peter Kubanda
Pavol Kubík
Michal Kurtý

Ing. Jana Flochová, PhD.

Neprítomní:

vypracoval: Lukáš Humený 6.11.2010

4.5 ZÁPISNICA 05

Piate stretnutie (8. týždeň semestra)
Dátum stretnutia : 8.11.2010

Program stretnutia:

1. Konzultácie (45 min)

- organizačné pokyny o odovzdávaní analýzy a špecifikácie
- prerozdělili sme si úlohy na vypracovanie teórie dokumentu nasledovne:
 - ✓ základy modelovania systémov - Jozef Ferenčík
 - ✓ petriho siete - Peter Kubanda
 - ✓ modelovacie nástroje - Lukáš Humený
 - ✓ špecifikácia a návrh - Tomáš Bartek
 - ✓ kontrola zápisníc, doplnenie a aktualizácia webovej stránky - Pavol Kubík
 - ✓ zoskupenie častí a vytvorenie kompletnej dokumentácie - Michal Kurtý
- dohodli sme sa na vypracovaní svojich častí do stredy 10.11.2010. Následne Michal Kurtý vytvorí dokument a pošle Ing. Flochovej a druhému tímu oponentov do 11.11.2010 14:00
- Ing. Flochová nám povedala základy o teórii zásobníkov (Queue theory)

2. Budúce stretnutie

- kontrola dokumentu
- návrh ďalšieho postupu
- hodnotenie druhého tímu

Prítomní:

Tomáš Bartek
Jozef Ferenčík
Lukáš Humený
Pavol Kubík
Michal Kurtý

Ing. Jana Flochová, PhD.

Neprítomní:

Peter Kubanda

vypracoval: Lukáš Humený 6.11.2010

4.6 ZÁPISNICA 06

Šieste stretnutie (9. týždeň semestra)
Dátum stretnutia : 15.11.2010

Program stretnutia:

1. Konzultácie (30 min)

- organizačné pokyny pri posudzovaní dokumentácie oponentského tímu prerozdělili sme si jednotlivé kapitoly dokumentu na napísanie posudku a to nasledovne:
 - ✓ Formálna stránka – Peter Kubanda
 - ✓ Úvod – Pavol Kubík
 - ✓ Analýza – Jozef Ferenčík
 - ✓ Špecifikácia – Tomáš Bartek
 - ✓ Návrh – Lukáš Humený
 - ✓ Zoskupenie častí a vytvorenie kompletného posudku - Michal Kurtý
- Kontrola dokumentu
 - ✓ dohodli sme sa na vypracovaní svojich častí do štvrtku 18.11.2010
 - ✓ následne Michal Kurtý vytvorí kompletný posudok a pošle ho Ing. Flochovej a druhému tímu oponentov do 19.11.2010, 14:00

2. Budúce stretnutie

- Oboznámenie sa s modelovacím prostredím nástrojov CPN Tools a Pipe2/3
Konzultácia dosiahnutých výsledkov, prípadne riešenie vzniknutých týmto nástrojmi

Prítomní:

Peter Kubanda
Pavol Kubík
Michal Kurtý

Ing. Jana Flochová, PhD.

Neprítomní:

Tomáš Bartek
Jozef Ferenčík
Lukáš Humený

vypracoval: Pavol Kubík 15.11.2010

Siedme stretnutie (10. týždeň semestra)

Dátum stretnutia : 22.11.2010

Program stretnutia:

1. Kontrola minulých bodov (20 min)

- posúdenie projektu oponentského tímu "**Vývoj vzorových aplikácií pre IPTV v prostredí Microsoft Mediaroom Framework**"
 - ✓ každý člen tímu posudzoval určitú časť oponentskej práce
 - Formálna stránka – Peter Kubanda
 - Úvod – Pavol Kubík
 - Analýza – Jozef Ferenčík
 - Špecifikácia – Tomáš Bartek
 - Návrh – Lukáš Humený
 - ✓ Zoskupenie častí a vytvorenie kompletného posudku - Michal Kurtý
- posudok sme odovzdali oponentskému tímu aj Ing. Flochovej
- zhodli sme sa, že daná téma má ambiciózne a perspektívne ciele s reálnym využitím v praxi, avšak sme zvedaví v akom rozsahu sa im to podarí naplniť pri finálnej implementácii

2. Konzultácie (30 min)

- s Ing. Flochovou sme rozoberali možnosti simulácií protokolov pomocou Petriho sietí v jednotlivých nástrojoch
 - uvažovali sme o vhodnosti použitia farebných a časových sietí a využitia podmienok v modeloch
- vyskúšali sme si vytvorenie jednoduchého modelu v simulátore PIPE2

3. Návrh konkrétneho modelu (30 min)

- navrhovali sme podrobný model správania OSPF na smerovači
 - modelovali sme jednotlivé stavy, ktorými prechádza smerovač pri vytváraní spojenie s inými OSPF smerovačmi

4. Budúce stretnutie

- dohodli sme sa, že do budúceho stretnutia si vyskúšame prácu na jednotlivých simulátoroch PIPE2 a CPN

- porovnáme funkcionality a výbere vhodnejšiu variantu

Prítomní:

Tomáš Bartek
Jozef Ferenčík
Lukáš Humený
Peter Kubanda
Pavol Kubík
Michal Kurtý

Ing. Jana Flochová, PhD.

Neprítomní:

vypracoval: Lukáš Humený 25.11.2010

4.8 ZÁPISNICA 08

ôsme stretnutie (11. týždeň semestra)
Dátum stretnutia : 29.11.2010

Program stretnutia:

1. Kontrola minulých bodov (10 min)

- priniesli sme vytvorený hrubý model a prekonzultovali sme správanie OSPF protokolu a prechodov cez jeho stavy na smerovačoch

2. Konzultácie (20 min)

- s Ing. Flochová nám vysvetlila príklad na modelovanie protokolu cez farebné Petriho siete. Rozoberali sme model z prednášky "Coloured Petri Nets" od Kurta Jensena
- prednáška zobrazovala podrobný postup pri návrhu sieťového protokolu. .
- danú prednášku sme si rozposlali na podrobnejšie doštudovanie problematiky

3. Budúce stretnutie

- prejdeme si danú prednášku a tvorbu modelov sieťových protokolov
- vytvorený hrubý model sa pokúsime vytvoriť v simulátore CPN Tools

Prítomní:

Tomáš Bartek
Jozef Ferenčík
Lukáš Humený
Peter Kubanda
Pavol Kubík
Michal Kurtý

Ing. Jana Flochová, PhD.

Neprítomní:

vypracoval: Lukáš Humený 2.12.2010

4.9 ZÁPISNICA 09

Deviate stretnutie (12. týždeň semestra)

Dátum stretnutia : 09.12.2010

Program stretnutia:

1. Kontrola minulých bodov (10 min)

- priniesli sme kompletný návrh modelu pre OSPF protokol a pokúsili sme sa vytvoriť jeho hrubý návrh v simulátore CPN Tools

2. Konzultácie (20 min)

- Ing. Flochovej sme predviedli hrubý návrh modelu, ktorý sme navrhli v simulátore CPN Tools
- Dospeli sme k záveru, že daný návrh má dobrý základ ako prototyp modelu, avšak bude potrebné doplniť jeho funkcionality.
- Dohodli sme sa na približnom dátume prezentácie projektu spolu s oponentským tímom a na predpokladanom obsahu prezentácie, pričom každý predvedie určitú časť z projektu, na ktorej pracoval

3. Budúce stretnutie

- Prezentácia prototypu pre oponentský tím
- Vytvorený prototyp v simulátore CPN Tools doplníme o funkcionality

Prítomní:

Tomáš Bartek
Jozef Ferenčík
Lukáš Humený
Peter Kubanda
Pavol Kubík
Michal Kurtý

Ing. Jana Flochová, PhD.

Neprítomní:

vypracoval: Pavol Kubík 9.12.2010

4.10 ZÁPISNICA 10

desiate stretnutie (2. týždeň letného semestra)

Dátum stretnutia : 24.02.2010

Program stretnutia:

1. Kontrola minulého semestra (10 min)

- zhodnotenie činnosti za minulý semester

2. Konzultácie (50 min)

- Michal Kurtý znázornil prototyp modelu v CPN
 - opísal možnosti modelu a nasledujúci postup a rozširovanie modelu
 - zistili sme hlavné nedostatky, ktoré treba doštudovať a doplniť, jednalo sa najmä o vytváranie prechodových podmienok v prostredí CPN Tools
- Ing. Flochová priniesla dizertačnú prácu o farebných petriho sieťach pre inšpiráciu nášho riešenia, obsahovala konkrétnu implementáciu sieťového protokolu
- Dohodli sme sa na stretávaní v stredu o 12:00 ak to bude možné v softvérovom štúdiu, ak nie ostáva stretávanie vo štvrtok o 12:00

3. Budúce stretnutie

Doštudovanie vývojového prostredia CPN Tools, hlavne implementácia prechodových podmienok

Prítomní:

Tomáš Bartek
Jozef Ferenčík
Lukáš Humený
Peter Kubanda
Pavol Kubík
Michal Kurtý
Michal Zapatický

Ing. Jana Flochová, PhD.

Neprítomní:

vypracoval: Lukáš Humený 26.02.2010

4.11 ZÁPISNICA 11

jedenáste stretnutie (3. týždeň letného semestra)
Dátum stretnutia : 02.03.2011

Program stretnutia:

1. Kontrola minulých bodov (10 min)

- Doštudovanie vývojového prostredia CPN Tools, hlavne implementácia prechodových podmienok

2. Konzultácie (50 min)

- Pokračovanie vo vývoji prototypu daného modelu v CPN Tools a riešenie implementácie podmienok. Konkrétne sa jednalo o podmienky, ako napr.:

```
if (status == DOWN) then status = INIT,
```

ktorá zmení stav (routra v prípade, že je DOWN) na stav INIT (inicializuje ho). Podmienky sa v CPN Tools riešia pomocou monitorov "Monitors", avšak ich implementácia nám nie je celkom jasná.

- Stretávanie tímu ostáva v stredu o 12:00 v miestnosti CD35

3. Budúce stretnutie

- Doštudovanie implementácie prechodových podmienok v prostredí CPN Tools

Prítomní:

Tomáš Bartek
Jozef Ferenčík
Peter Kubanda
Pavol Kubík
Michal Zapatický

Ing. Jana Flochová, PhD.

Neprítomní:

Lukáš Humený
Michal Kurtý

vypracoval: Pavol Kubík 03.03.2011

4.12 ZÁPISNICA 12

dvanáste stretnutie (4. týždeň letného semestra)

Dátum stretnutia : 09.03.2011

Program stretnutia:

1. Kontrola minulých bodov (10 min)

- zhodnotenie naštudovania syntaxe CPN (najmä podmienky a úprava prenášaných dát)
- Ing. Flochová rieši inštaláciu CPN na niektorý počítač v softvérovom štúdiu

2. Kolektívna činnosť tímu (1,5 hod)

- doštudovali sme si tvorbu podmienených simulácií v CPN Tools
 - vytvorili sme konkrétny návrh čiastkového modelu v CPN
 - vytvorené podmienkové hrany, kontrola parametrov prenášaných dát, simulácia náhodnosti prenosu (chyby)
- hľadanie spôsobu ako meniť konkrétne polia pri komunikácii dvoch OSPF uzlov

3. Budúce stretnutie

- študovanie vývojového prostredia CPN Tools, hlavne implementácia zmeny jednotlivých polí v záznamoch
- snaha o dosiahnutie hierarchického modelu v CPN

Prítomní:

Tomáš Bartek
Michal Kurtý
Jozef Ferenčík
Lukáš Humený
Peter Kubanda
Michal Zapatický

Ing. Jana Flochová, PhD.

Neprítomní:

Pavol Kubík

vypracoval: Lukáš Humený 10.03.2011

4.13 ZÁPISNICA 13

trináste stretnutie (5. týždeň letného semestra)
Dátum stretnutia : 16.03.2011

Program stretnutia:

1. Kontrola minulých bodov (10 min)

- zhodnotenie naštudovania syntaxe CPN (najmä implementácia zmeny jednotlivých polí v záznamoch a dosiahnutie hierarchického modelu v CPN)

2. Kolektívna činnosť tímu (1,5 hod)

- doštudovali sme si tvorbu implementáciu zmien polí v záznamoch v CPN Tools
- vytvorili sme konkrétny návrh čiastkového modelu v CPN
- vytvorili sme čiastočný hierarchický model v CPN
- vytvorili sme prechody medzi routami do INIT a 2WAY stavov

3. Budúce stretnutie

- pokračovanie v rozvíjaní a dopĺňaní hierarchického modelu v CPN

Prítomní:

Jozef Ferenčík
Lukáš Humený
Peter Kubanda

Pavol Kubík
Michal Kurtý
Michal Zapatický

Ing. Jana Flochová, PhD.

Neprítomní:

Tomáš Bartek

vypracoval: Pavol Kubík 18.03.2011

4.14 ZÁPISNICA 14

štrnásť stretnutie (6. týždeň letného semestra)
Dátum stretnutia : 23.03.2011

Program stretnutia:

1. Kontrola minulých bodov (10 min)

- Kontrola hierarchického modelu v CPN z minulého týždňa

2. Kolektívna činnosť tímu (1,5 hod)

- pokračovali sme v tvorbe modelu OSPF v CPN Tools
 - snažili sme sa vytvoriť hierarchický model kvôli lepšej prehľadnosti
 - doplnili sme DR a DBR informácie v Hello paketoch
- Ing. Flochová sa pokúsi priviesť študenta so sieťovými znalosťami ku konzultácii nášho modelu

3. Budúce stretnutie

- pokračovanie v rozvíjaní a dopĺňaní hierarchického modelu v CPN

Prítomní:

Tomáš Bartek
Jozef Ferenčík
Lukáš Humený
Peter Kubanda
Pavol Kubík
Michal Kurtý
Michal Zapatický

Ing. Jana Flochová, PhD.

Neprítomní:

vypracoval: Pavol Kubík 24.03.2011

4.15 ZÁPISNICA 15

pätnáste stretnutie (7. týždeň letného semestra)

Dátum stretnutia : 30.03.2011

Program stretnutia:

1. Kontrola minulých bodov (10 min)

- Kontrola hierarchického modelu v CPN z minulého týždňa

2. Kolektívna činnosť tímu (1,5 hod)

- Pokračovali sme v tvorbe modelu OSPF v CPN Tools
 - ✓ rozhodli sme sa tvorenie hierarchického modelu odložiť na neskôr, nakoľko v aktuálnom modeli sa zatiaľ lepšie orientujeme
 - ✓ pokračovali sme v implementácii informácie DR a DBR v Hello paketoch
- Ing. Flochová sa na budúce stretnutie pokúsi priviesť Ing. Petra Trúchleho so sieťovými znalosťami ku konzultácii nášho modelu

3. Budúce stretnutie

- pokračovanie v rozvíjaní a dopĺňaní modelu OSPF v CPN Tools

Prítomní:

Tomáš Bartek
Jozef Ferenčík
Lukáš Humený
Peter Kubanda
Pavol Kubík
Michal Kurtý
Michal Zapatický

Ing. Jana Flochová, PhD.

Neprítomní:

vypracoval: Pavol Kubík 30.03.2011

4.16 ZÁPISNICA 16

šestnásť stretnutie (8. týždeň letného semestra)

Dátum stretnutia : 06.04.2011

Program stretnutia:

1. Kontrola minulých bodov (10 min)

- Kontrola hierarchického modelu v CPN z minulého týždňa

2. Kolektívna činnosť tímu (1,5 hod)

- Pokračovali sme v tvorbe modelu OSPF v CPN Tools
 - pokračovali sme v implementácii informácie DR a DBR v Hello paketoch a odstránili sme menšie problémy v implementácii susedných paketov
- Ing. Peter Trúchly prišiel konzultovať náš OSPF model (sieťarskeho hľadiska)
 - po konzultácii sa ešte pokúsime doplniť časovú zložku a stratovosť paketov (štatistika) do nášho OSPF modelu, prípadne ešte zlyhanie niektorého smerovača

3. Budúce stretnutie

- pokračovanie v rozvíjaní a dopĺňaní modelu OSPF v CPN Tools

Prítomní:

Tomáš Bartek
Jozef Ferenčík
Peter Kubanda
Pavol Kubík
Michal Kurtý
Michal Zapatický

Ing. Jana Flochová, PhD.

Neprítomní:

Lukáš Humený

vypracoval: Pavol Kubík 06.04.2011

4.17 ZÁPISNICA 17

sedemnásť stretnutie (9. týždeň letného semestra)

Dátum stretnutia : 13.04.2011

Program stretnutia:

1. Kontrola minulých bodov (10 min)

- Kontrola hierarchického modelu v CPN z minulého týždňa
- Implementovali sme štatistické vyhodnotenie paketov (stratovosť)

2. Kolektívna činnosť tímu (1,5 hod)

- Pokračovali sme v tvorbe modelu OSPF v CPN Tools
- pokračovali sme v implementácii informácie DR a DBR v Hello paketoch
- dokončili sme hierarchickú architektúru modelu a kompaktnosť medzi jednotlivými spájanými celkami

3. Budúce stretnutie

- pokračovanie v rozvíjaní a dopĺňaní modelu OSPF v CPN Tools
- redukcia niektorých častí modelu z dôvodu neprehľadnosti
- plánovaná chybovosť (zlyhanie) jedného smerovača

Prítomní:

Tomáš Bartek
Jozef Ferenčík
Pavol Kubík
Michal Kurtý
Michal Zapatický

Ing. Jana Flochová, PhD.

Nepřítomní:

Lukáš Humený
Peter Kubanda

vypracoval: Pavol Kubík 13.04.2011

4.18 ZÁPISNICA 18

osemnásť stretnutie (10. týždeň letného semestra)

Dátum stretnutia : 20.04.2011

Program stretnutia:

1. Kontrola minulých bodov (10 min)

- Plánovaná chybovosť (zlyhanie jedného smerovača)
 - Čiastočne sa nám podarilo implementovať zlyhanie smerovača v rámci OSPF modelu
 - Zredukovali sme niektoré časti modelu kvôli lepšej prehľadnosti

2. Kolektívna činnosť tímu (1,5 hod)

- Pokračovali sme v tvorbe modelu OSPF v CPN Tools
 - Čiastočne sa nám podarilo implementovať stratovosť paketov v modelovanej sieti
 - vylepšili sme hierarchickú architektúru modelu a kompaktnosť medzi jednotlivými spájanými celkami

3. Budúce stretnutie

- Pokračovanie v rozvíjaní a dopĺňaní modelu OSPF v CPN Tools
 - Vyladenie a dokončenie hierarchie v OSPF modeli
 - Dokončenie stratovosti paketov v sieti

Prítomní:

Jozef Ferenčík
Lukáš Humený
Peter Kubanda
Pavol Kubík
Michal Kurtý
Michal Zapatický

Neprítomní:

Ing. Jana Flochová, PhD.
Tomáš Bartek

vypracoval: Pavol Kubík 20.04.2011

4.19 ZÁPISNICA 19

devätnáste stretnutie (11. týždeň letného semestra)
Dátum stretnutia : 27.04.2011

Program stretnutia:

1. Kontrola minulých bodov (10 min)

- Podarilo sa nám dokončiť model OSPF v CPN Tools do takej miery, že máme dokončenú stratovosť paketov, doplnenú informáciu DR a DBR v paketoch, a kompletne hierarchický model

2. Kolektívna činnosť tímu (1,5 hod)

- Dokončovanie modelu OSPF v CPN Tools, úprava malých nedostatkov

3. Budúce stretnutie

- Dokončenie poslednej fázy tvorby nášho OSPF modelu v CPN Tools a kontrola jeho funkčnosti, tvorba kompletnej dokumentácie, aktualizácia webu

Prítomní:

Tomáš Bartek
Jozef Ferenčík
Lukáš Humený
Peter Kubanda
Michal Kurtý

Neprítomní:

Ing. Jana Flochová, PhD.
Pavol Kubík
Michal Zapatický

vypracoval: Pavol Kubík 27.04.2011

4.20 ZÁPISNICA 20

dvadsiate stretnutie (12. týždeň letného semestra)

Dátum stretnutia : 03.05.2011

Program stretnutia:

1. Kontrola minulých bodov (10 min)

- Podarilo sa nám dokončiť model do takej miery ako sme sa dohodli s vedúcou projektu Ing. Flochovou, upravili sme pár nedostatkov

2. Kolektívna činnosť tímu (1,5 hod)

- Kontrola a testovanie výsledného modelu OSPF v CPN Tools, dokončili sme dokumentáciu za letný semester, takisto aj dokumentáciu riadenia a všetky

zápisnice, aktualizovali sme webovú stránku

3. Budúce stretnutie

- Príprava tímu na obhajobu nášho projektu, písanie posudku pre oponentský tím

Prítomní:

Tomáš Bartek
Jozef Ferenčík
Lukáš Humený
Peter Kubanda
Pavol Kubík
Michal Kurtý
Michal Zapatický

Ing. Jana Flochová, PhD

Neprítomní:

vypracoval: Pavol Kubík 04.05.2011