# Laboratórne cvičenie 5 – Prepojovanie paketov

**Zadanie**

 Jedna z technológií na prenos dát, ktorá sa používa v počítačových sieťach sa nazýva “prepojovaním paketov“. Dáta sú vtedy posielané postupne po menších častiach (paketoch). Smerovanie v uzloch zabezpečujú prepínače (switch, router).



Naša topológia sa sa skladá z troch uzlov. Uzol 1 je spojený s uzlom 3 a uzol 3 je prepojený s uzlom 2. Každý z nich obsahuje v sebe okrem modulov potrebných na smerovanie paketov ešte cieľ a zdroj. Rýchlosť kabeláže je 128kb/s. Priemerná veľkosť paketov prechádzajúcich cez naše uzly je 400B teda 3200b. Čas smerovania je nulová. Meraním sme zistili intenzity tokov, ktoré z okolia paketovej siete vstupujú do uzlov 1, 2, 3. Intenzity sú porade 40, 20, 30 paketov za sekundu. Pravdepodobnosti smerovania paketov v uzloch siete udáva nasledujúca tabuľka:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ktorý\Kam** | **do Uzla1** | **do Uzla2** | **do Uzla3** | **Ciel** |
| **z Uzla1** | 0 | 0 | 0.25 | 0.75 |
| **z Uzla 2** | 0 | 0 | 0.5 | 0.5 |
| **z Uzla 3** | 0.25 | 0.5 | 0 | 0.25 |

## Ciele

* Aká je intenzita tokov na jednotlivých linkách siete? Vyhodnotiť výsledky o toku, ktorý tečie do cieľa
* Aké je stredné oneskorenie na jednotlivých linkách siete?
* Koľko paketov priemerne čaká vo fronte pred linkou z uzla 1 do uzla 3?
* Za aký čas prejde paket sieťou (určte priemernú hodnotu)?

## Príprava

* Vytvorenie potrebných Process modelov
* Vytvorenie Node modelov
* Vytvorenie modelu linky

Na vytvorenie našej siete musíme najskôr vytvoriť uzly. Uzly 1 a 2 budú mať rovnaký Node model. Na jeho vytvorenie budeme potrebovať nasledovné moduly z Node Editora: jeden vysielač, jeden prijímač a tri procesory. Jeden procesor bude pakety generovať (Zdroj), druhý bude cieľom pre pakety(Cieľ), tretí bude rozhodovať o smerovaní prevádzky (Smerovač) a vysielač s prijímačom budú slúžiť na spojenie s ostatnými uzlami. Uzol tri bude na rozdiel od uzlov 1 a 2 obsahovať dva vysielače a dva prijímače.

Aby sme splnili počet paketov za sekundu generovaných v zdrojoch použijeme konštantnú veľkosť pre generovanie medzier medzi paketmi.

Začneme vytvorením node modelov.

1. V novom okne vyberieme z menu **File** > **New** … a vyberieme **Node Model**. Potvrdíme **OK**.
2. Použitím vhodných nástrojov umiestnime na plochu tri procesory a jeden front tak ako sú rozmiestnené na obrázku. Pre prijímač a vysielač použijeme ikonky v krúžku.



1. Pre každý objekt nadstavíme meno tak ako je na obrázku.
2. Objekty pospájame tokmi tak ako je ukázané na obrázku v poradí

Smerovac -> Vysielač

Smerovac -> Ciel

Prijímač -> Smerovac

Zdroj -> Smerovac

Pre **Zdroj** nadstavíme nasledovné atribúty:

1. Zmeníme **process model** atribút na **simple\_source**.
2. Nadstavíme **Packet Interarrival Time** na e**xponential (0.025)**
3. Pre **Packet Size** zadajte **exponential (3200)**
4. **OK.**

Konečné nadstavenie objektu Zdroj by malo vyzerať nasledovne:



Pokračujeme nadstavením atribútov objektu **Vysielač**:

1. Klikneme pravým tlačidlom na myši na ikonu procesora a vyberieme **Edit Attributes**
2. Klikneme na atribút **channel** v stĺpci Value – **(...)**.
3. V novo vyskočenom okne zmeníme **data rate (dbs)** na **128000**
4. Klikneme 2x **OK.**

Tieto kroky zopakujeme aj pre objekt prijímač.

Ďalej je potrebné správne nadstaviť rozhrania uzlov, na typ **fixed** a uložiť súbor:

1. Z hlavného menu vyberieme: **Interfaces > Node Interfaces**.
2. V okienku **Node types**, zmeníme **Supported** na **no** pre mobilné a satelitné typy.
3. V tabuľke Attributes zmeníme všetkým atribútom ~~nadstavíme~~ **Status** na **Hidden**
4. **OK. File > Save** meno **<inicialky>\_ LC5\_Prepojovanie\_paketov\_node1** a **Save**. EDITOR NEZAVRIEME!

Smerovač, ktorý bude riadiť s akou pravdepodobnosťou bude paket poslaný ktorou cestou, musíme urobiť sami. Každý uzol bude mať vlastný.

## Vytvorenie objektu v process editore

**Process model** obsahuje:

* **Stavy** - aktuálny stav modulu aj s hodnotami atribútov. Modul môže napríklad čakať na udalosť príchod paketa.
* **Cesty** – reakcia na udalosť zmenou stavu

Poznáme dva typy stavov. Rozlišujeme ich na základe toho, či sa po vykonaní kódu môže riadenie presunúť do iného stavu, alebo je potrebné čakať na nejakú špecifickú udalosť.

Prvý typ sa nazýva „unforced“ (zelený) a druhý „forced“ (červený).



Sú tri miesta kde sa dá dopisovať Proto-C kód:

* **Pri vstupe do stavu(Enter)** – kód je vykonaný, keď sa modul presunie do stavu
* **Pri výstupe zo stavu(Exit)** – kód sa vykoná pri opustení stavu
* **Do ciest** – kód sa vykoná ako reakcia na udalosť



Náš smerovač paketov bude obsahovať dva stavy – Inicializačný a neaktívny. Medzi nimi sa budú nachádzať tri cesty : Inicializačný-Neaktívny a dvakrát Neaktívny-Neaktívny.

Teraz pristúpime ku konkrétnym krokom pri vytváraní modulu Smerovač:

1. V okne node editora vyberieme **File** > **New** … a vyberieme **Process Model**. Klikneme **OK**.
2. Označíme **Create State**  a dvakrát klikneme na pracovnú plochu Process Model Editora.
3. Klikneme pravým tlačidlom na stavy a výberom **Set Name** nadstavíme mená postupne na **Inicializacny** a **Neaktivny**
4. Zmeníme stav **Inicializacny** pravým kliknutím myšky na **Make State Forced** (vyfarbí sa na zeleno)



V stave **Inicializacny** si nadstavíme premennú, ktorú budeme neskôr potrebovať:

1. Dvakrát klikneme na hornú polovicu stavu **Inicializacny**
2. Vložíme nasledujúci kód pre inicializovanie premennej **pomocna**:

*pomocna = 0 ;*

1. Uložíme **CTRL+S** a zavrieme **CTRL+W**.

Teraz spravíme cesty medzi stavmi. Sú dva typy ciest – s/bez podmienky. Ak cesta obsahuje podmienku, hodnota musí byť true ak sa po nej má riadenie presunúť zo zdrojového stavu do iného.

1. Klikneme na **Create Transition .**
2. Nakreslime cestu kliknutím na **Inicializacny** a potom na **Neaktivny**.
3. Klikneme na **Neaktivny,** spravíme medziklik aby cesta vytvorila brucho a opäť klikneme na **Neaktivny**. Toto vykonáme dvakrát.
4. Klikneme pravým na cestu z **Neaktivny** do **Neaktivny** a vyberieme **Edit Attributes.**
5. Zmeníme hodnotu **condition**  na **PRICHOD** (všetky veľké) a klikneme **OK.**

(pokiaľ nebude hodnota v konštante PRICHOD „true“ tak sa riadenie nepresunie)

1. Klikneme na **executive** a napíšeme **Cesta()**.

(pokiaľ bude v podmienka splnená vykoná sa obsah funkcie **Cesta**)

1. Klikneme pravým na druhú cestu z **Neaktivny** do **Neaktivny** a vyberieme **Edit Attributes.**
2. Zmeníme hodnotu **condition**  na **default**(všetky malé) a klikneme **OK.**



Dôvod prečo sú dve cesty v stave Neaktívny do samého seba je nasledovný. Počas prebiehania simulácie program obsluhuje kalendár udalostí ktoré chcú prebrať riadenie. Keď sa udalosť dostane na vrchol kalendára udalostí - nastane prerušenie. Prerušenia sa dostanú do modulu a to aktivuje Process model nášho modulu.

Pokiaľ bude podmienka PRICHOD splnená príde prerušenie spôsobené príchodom paketa. Ak však nastane iné prerušenie musí existovať cesta, ktorou sa môže náš model pohnúť. **Default** cesta práve zahrnuje tieto typy prerušení.

Deklarovanie premenných:

1. Klikneme na **Edit Header Block **
2. Napíšeme do novootvoreného okna tento kód:

#define PRICHOD (op\_intrpt\_type () == OPC\_INTRPT\_STRM)

1. Vyberieme  **File > Commit**  z menu okna.
2. Vyberieme **File > Close** pre zavretie.

Táto **PRICHOD** podmienka definovaná vyššie kontroluje prerušenie spôsobené príchodom paketa. Porovná hodnotu vrátenú procedúrou **op\_intrpt\_type()** s **OPC\_INTRPT\_STRM** konštantnou. Ak bude porovnanie pravdivé bude to znamenať, že prerušenie je to na ktoré čakáme.

Ďalším krokom je deklarácia pomocnej premennej. Využijeme ju pri generovaní pravdepodobnosti smerovania paketa.

1. Klikneme na **Edit State Variables **.
2. Vložíme nasledujúce údaje do otvoreného okna:



1. Klikneme **OK**.

Deklarovanie funkcie Cesta:

1. Klikneme na **Edit Function Block**



1. Napíšeme do novootvoreného okna tento kód:

#include <stdlib.h> /\* srand, rand \*/

*static void Cesta(void)*

 *{*

 *Packet \*pkptr;*

 *FIN(Cesta());*

 *pkptr = op\_pk\_get (op\_intrpt\_strm ());*

 *pomocna = rand() % 100 + 1;*

*if (pomocna < 25) op\_pk\_send\_forced (pkptr, 0);*

 *//paket stream [0] do vysielaca (uzol3)*

 *else op\_pk\_send\_forced (pkptr, 1);*

 *//paket stream [1] do Ciela*

 *FOUT;*

 *}*

1. Vyberieme  **File > Commit**  z menu okna.
2. Vyberieme **File > Close** pre zavretie.

Na začiatku funkcie si uložíme do premennej pkptr smerník na paket, ktorý spôsobil prerušenie. Do premennej pomocna si vygenerujeme číslo pomocou ktorého potom pošleme paket jedným, alebo druhým tokom. Argumentmi funkcie op\_pk\_send\_forced je paket, ktorý chceme odoslať a číslo toku (streamu), ktorým ho chceme odoslať.

Ak chceme atribúty dostupné aj na úrovni Node Editora musíme upraviť Process Interfaces.

1. Vyberieme **Interfaces > Process Interfaces**. Otvorí sa nové dialógové okno.
2. Zmeníme hodnotu **Initial Value** pre **begsim intrpt** attribút na **enabled**.

Spôsobí to, že simulátor doručí prerušenie do proces modelu na začiatku simulácie.

1. Skontrolujeme, že všetky hodnoty pre **endsim intrpt, failure intrpts, intrpt interval, recovery intrpts,** a **super priority** je hodnota nadstavená na **disabled**.
2. Pre všetky atribúty nadstavíme hodnotu **Status** na **hidden**, takže tieto atribúty zmiznú v node editore.
3. Klikneme **OK**.

Nasleduje kompilácia modelu:

1. Klikneme na **Compile Process Model **
2. Uložíme náš model ako **<iniciálky>\_LC5\_process\_m\_uzol1** do vašej zložky op\_models.
3. Po skompilovaní klikneme na **Close**.

Keďže máme tri uzly pri ktorých sú objekty Smerovač veľmi podobné pripravíme si teraz aj zvyšné dva. Najskôr uložíme náš model **<iniciálky>\_LC5\_process\_m\_uzol1** pod iným menom.

1. **File > Save as <iniciálky>\_LC5\_process\_m\_uzol2**
2. Otvoríme **Function block **
3. V 7. riadku zmeníme podmienku na **(pomocna < 50)**
4. Uložíme**(CTRL+S)** a zavrieme**(CTRL+W)**
5. Klikneme **File > Save as** a znova uložíme  **<iniciálky>\_LC5\_process\_m\_uzol3**
6. Otvoríme **Function block**
7. A premažeme kód nasledujúcim

#include <stdlib.h> /\* srand, rand \*/

*static void Cesta(void)*

 *{*

 *Packet \*pkptr;*

 *FIN(Cesta());*

 *pkptr = op\_pk\_get (op\_intrpt\_strm ());*

 *pomocna = rand() % 100 + 1;*

*if (pomocna < 25)*

 *op\_pk\_send\_forced (pkptr, 0);*

 *//paket stream [0] do Vysielaca*

*else*

 *if ((pomocna > 25) && (pomocna < 50))*

 *op\_pk\_send\_forced (pkptr, 1);*

 *//paket stream [1] do Ciela*

 *else*

 *op\_pk\_send\_forced (pkptr, 2);*

 *//paket stream [2] do vysielaca2*

 *FOUT;*

 *}*

1. Uložíme**(CTRL+S)** a zavrieme**(CTRL+W)**
2. Zavrieme aj Process editor **(CTRL+W)**

Vyššie uvedený kód je len obohatený o možnosť troch ciest.

Teraz, keď sme si vytvorili všetky potrebné moduly, môžeme dorobiť node model prvého a ostatných uzlov.

1. Z proces editora vyberieme **Windows>Node model> <inicialky>\_ LC5\_Prepojovanie\_paketov\_node1**
2. Pre Smerovač nadstavíme atribút **process model** na **<iniciálky>\_LC5\_process\_m\_uzol1**
3. Uložíme náš prvý uzol **(CTRL+S)**

Urobíme druhý uzol:

1. Klikneme **File > Save as** a uložíme  **<inicialky>\_LC5\_Prepojovanie\_paketov\_node2**
2. Klikneme pravým tlačidlom na **Zdroj** a vyberieme **Edit Attributes**
3. Pre objekt **Zdroj** nadstavíme **Packet Interarrival Time** na **exponential (0.05)**
4. Pre **Smerovač** nadstavíme atribút **process model** na **<iniciálky>\_LC5\_process\_m\_uzol2**
5. Uložíme náš druhý uzol **(CTRL+S)**

Tretí uzol:

1. Klikneme **File > Save as** a uložíme  **<inicialky>\_LC5\_Prepojovanie\_paketov\_node3**
2. Klikneme pravým tlačidlom na **Zdroj** a vyberieme **Edit Attributes**
3. Pre objekt **Zdroj** nadstavíme **Packet Interarrival Time** na **exponential (0.03333)**
4. Pre Smerovač nadstavíme atribút **process model** na **<iniciálky>\_LC5\_process\_m\_uzol3**
5. Umiestnime podľa obrázku nižšie **Point-to-point Receiver**
6. Umiestnime **Point-to-point Transmitter**
7. Pomenujeme ich ako je na obrázku
8. Vytvoríme packet streami ako na obrázku
9. Klikneme pravým tlačidlom na myši na ikonu **Prijimac2** a vyberieme **Edit Attributes**
10. Klikneme na atribút **channel** v stĺpci Value – **(...)**.
11. V novo vyskočenom okne zmeníme **data rate (dbs)** na **128000**
12. Klikneme 2x **OK.**
13. Kroky 7-10 zopakujeme aj pre **Vysielač2**.
14. Uložíme**(CTRL+S)** a zavrieme**(CTRL+W)**



Nasledujúcimi krokmi si pripravíme **model linky**, ktorú použijeme na prepojenie uzlov:

1. Klikneme na **File > New...**  a vyberieme **Link Model**. **OK**.
2. V okne **Supported link types**, zmeníme hodnotu **Supported** na **no** pre všetky riadky okrem **ptdup**. Týmto sme si vybrali poit-to-point spojenie ktoré bude obojsmerné (duplex).



1. V okne **Attributes** nadstavíme **data rate** na **128000** (b/s)
2. Nadstavíme **propdel model** na **dpt propdel** (point-to-point propagation delay) - oneskorenie pri šírení bolo pôvodne nadstavené pre zbernicu – bus)
3. Nadstavíme **txdel model** na **dpt txdel** (point-to-point transmission delay)
* oneskorenie spôsobené rýchlosťou linky treba tiež nadstaviť na ptp.

Rozdiel medzi oneskorením pri šírení (propagation) a oneskorením pri prenose (transmission) je, že transmission delay nemá nič spoločné so vzdialenosťou medzi dvoma uzlami. Vypočítame ho vydelením počtu bitov rýchlosťou linky (data rate). Propagation delay je odvodené od rýchlosti šírenia signálu a vzdialenosti medzi uzlami. Tak vypočítam čas potrebný na prechod signálu po fyzickom médiu.

1. Nadstavíme **ecc model**  to  **ecc\_zero\_err**
2. Nadstavíme **error model**  to **NONE**
3. Vyberieme **File > Declare External Files...**
4. (objaví sa Declare External Files okno)
5. Vyhľadáme **Link Delay** a zaznačíme ho. Okno zavrieme.
6. Vyberieme **File > Save** a nazveme **<inicialky>\_LC5\_link\_m**
7. Zavrieme link editor

**Vytvorenie siete**

Teraz máme vytvorené všetko potrebné aby sme boli schopný namodelovať našu sieť.

1. **File >New...** ponecháme **Projct** a klikneme **OK.** Meno nového projektu **<inicialky>\_LC5\_Prepojovanie\_paketov.** Pre **Scenario name** necháme **scenario 1.** Potvrdíme **OK**.
2. Pri Startup Wizard stlačíme **Quit**

Na pracovnú plochu umiestnime naše tri uzly a prepojíme ich linkou ktorú sme vytvorili. Uzly uložíme tak aby ich pozície odpovedali obrázku na obrázku nižšie.

1. Z panelu nástrojov si vyberieme **Open Object Palette:**
2. Do poľa **Search by name** napíšeme **<iniciálky>\_LC5** a stlačíme enter.
3. Po nájdení našich uzlov umiestnime každý na pracovnú plochu project editora a pomenujeme ich tak ako sú pomenované na obrázku.
4. Teraz do poľa **Search by name** napíšeme meno nášho link modelu - **<inicialky>\_LC5\_link\_m**
5. Po nájdení spojíme v tomto poradí **uzol1 –> uzol3** a **uzol3 –> uzol2**
6. Paletu zavrieme kliknutím na **Close**.



Linku ktorou sme pospájali uzli treba osobitne pripojiť na konkrétne prijímače a vysielače:

1. Klikneme pravým tlačidlom myšky na linku **uzol1 <–> uzol3** a vyberieme **Edit attributes.**



Pokiaľ nie sú hodnoty nadstavené ako na obrázku musíme nadstaviť na obidvoch linkách pre **NONE** a pospájať ich nasledujúcimi krokmi:

1. Pre **transmitter a**nadstavíme **Uzol1.Vysielac**
2. Pre **receiver a**nadstavíme **Uzol1.Prijimac**
3. Pre **transmitter b**nadstavíme **Uzol3.Vysielac**
4. Pre **receiver a**nadstavíme **Uzol3.Prijimac**
5. Klikneme **OK**
6. Pre linku **Uzol2 <-> Uzol3** a vyberieme **Edit attributes.**
7. Pre **transmitter a**nadstavíme **Uzol2.Vysielac**
8. Pre **receiver a**nadstavíme **Uzol2.Prijimac**
9. Pre **transmitter b**nadstavíme **Uzol3.Vysielac2**
10. Pre **receiver a**nadstavíme **Uzol3.Prijimac2**
11. Klikneme **OK**

Skontrolujeme hodnoty aj na druhej linke podľa obrázku:



**Zbieranie výsledkov**

Voľbu **Choose Individual DES Statistics** je možné nájsť kliknutím pravého tlačidla myšky na tri miesta:

* Pracovnú plochu
* Uzol
* Linku

Pre skrátenie vypisovania budeme používať iba DES plochy, DES uzla xy a DES linky xy.

Teraz ponadstavujeme ktoré štatistiky sa budú mať zbierať aby sme mohli odpovedať na zadané úlohy:

* Aká je intenzita tokov na jednotlivých linkách siete? Vyhodnotiť výsledky o toku ktorý tečie do cieľa.
1. V DES **linky** **uzol1 <–> uzol3** zaznačíme  **point-to-point >** **throughput (pakets/sec) 🡪** a **>** **throughput (pakets/sec) 🡨**
2. Zopakujeme pre linku **Uzol2 <-> Uzol3**
3. V DES všetkých troch **uzlov** zaznačíme **Module Statistics > Ciel > Traffic Sink >** **Traffic Received (packets/sec)**
* Aké je stredné oneskorenie na jednotlivých linkách siete?
1. V DES **linky** **uzol1 <–> uzol3** zaznačíme **point-to-point >** **queuing delay 🡪 (sec)** a **queuing delay (sec) 🡨**
2. Zopakujeme pre linku **Uzol2 <-> Uzol3**
* Koľko paketov priemerne čaká vo fronte pred linkou z uzla 1 do uzla 3?
1. V DES **Uzla1** zaznačíme **Module Statistics > Vysielac.channel[0] > point-to-point transmitter > queue size (packets)**
* Za aký čas prejde paket sieťou (určiť priemernú hodnotu)?
1. V DES plochy zaškrtneme **Global Statistics > Traffic Sink > End-to-End Delay (second)**

Stlačíme **CTRL+S, OK** a **CTRL+R** – skontrolujeme či je **Duration** nadstavené na **10 hour(s)** a stlačíme **Run.**

**Prezeranie výsledkov**

Keďže toto tiež nie je prvý krát čo ideme prezerať výsledky tiež si zavedieme malé zjednodušenie. Výsledky sa dajú prezerať v dvoch oknách, ktoré otvoríme z Project editora nasledujúcim spôsobom:

* Result browser : **DES > result > view result...**
* Select Statistic for Top Result : **DES > result > Find Top Statistics...**

Okno Select Statistic for Top Result - budeme skrátene volať Top Result. Na zobrazenie vybranej štatistiky vždy použijeme tlačidlo **Find Top Result.** Na prezeranie budeme používať Top Result pretože je v ňom vyjadrená štatistika aj číselne.

Intenzita tokov na jednotlivých linkách siete:

* **Top result** v **Link Statistics > point-to-point > throughput (pakcet/sec)**





 Výsledky o toku ktorý tečie do cieľov:

* **Top result** v **Node Statistics > Traffic Sink > Traffic Received (bit/sec)**





Stredné oneskorenie na jednotlivých linkách siete:

* **Top result** v **Link Statistics > point-to-point > queuing delay (sec)**





Počet paketov priemerne čakajúcich vo fronte pred linkou z uzla 1 do uzla 3:

* **Top Statistics** v **Node Statistics > point-to-point transmitter > queue size (packets)**





Čas za ktorý paket prejde sieťou:

* **Result Browser** v liste **Des Graphs** rozbalíme **Global Statistics > Traffic Sink > End-to-End Delay (seconds)**
* V pravo dole v okne **Presentation** zmeníme **As Is** na **average**



Priemerná hodnota sa po minúte ustálila na 0.223

**Záver**

Pokiaľ porovnáme výsledky s matematickým modelom vidíme, že priemerný počet paketov, ktoré prejdú jednotlivými tokmi vyšli rovnako s minimálnymi rozdielmi.

|  |  |  |
| --- | --- | --- |
| Intenzita toku linky | Matematický model | Simulačný model |
| z uzla 3 do uzla 2 | 36,36 | 37,4 |
| z uzla 2 do uzla 3 | 28,18 | 28,03 |
| z uzla 3 do uzla 1 | 18,18 | 17,15 |
| z uzla 1 do uzla 3 | 14,54 | 13,76 |

Rovnako vyšiel aj stredný počet paketov, ktoré odchádzajú zo siete do cieľov vyšli veľmi podobne:

|  |  |  |
| --- | --- | --- |
| Intenzita toku linky | Matematický model | Simulačný model |
| z uzla 1 von zo siete | 43,64 | 43,42 |
| z uzla 2 von zo siete | 28,18 | 29,28 |
| z uzla 3 von zo siete | 18,18 | 17,22 |

Priemerný čas, za ktorý sú pakety prenesené jednotlivými linkami, a za ktorý prejde paket sieťou vyšiel podobne, no v simulačnom modeli sú tieto časy väčšie:

|  |  |  |
| --- | --- | --- |
|  | Matematický model | Simulačný model |
| z uzla 3 do uzla 2 | 0,2176 | 0,4412 |
| z uzla 2 do uzla 3 | 0,0782 | 0,0695 |
| z uzla 3 do uzla 1 | 0,0439 | 0,04477 |
| z uzla 1 do uzla 3 | 0,0379 | 0,03803 |

Počet paketov priemerne čakajúcich vo fronte pred linkou z uzla 1 do uzla 3 a

stredné oneskorenie, za ktoré prejde paket sieťou:

|  |  |  |
| --- | --- | --- |
|  | Matematický model | Simulačný model |
| Front z uzla 1 do uzla 3 | 0,2176 | 0,4412 |
| Stredné oneskorenie | 0,0966 | 0,2231 |

Napriek tomu, že priemerný čas za ktorý sú pakety prenesené jednotlivými linkami a čas za ktorý prejde paket sieťou sú mierne väčšie, môžeme skonštatovať, že matematický model odpovedá simulačnému modelu.