

Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

## BAKALÁŘSKÁ PRÁCE



Michal Bečka

### **Simulace systému pro směřování dlouhých zpráv**

Katedra aplikované matematiky

Vedoucí bakalářské práce: Prof. RNDr. Luděk Kučera, DrSc.

Studijní program: Informatika, správa počítačových systémů

2007

Rád bych poděkoval všem, co mě podporovali během psaní této bakalářské práce, především za jejich trpělivost při dlouhých dnech mého programování a testování. Také bych rád poděkoval svému vedoucímu Prof. RNDr. Luďku Kučerovi, DrSc. Za výběr tématu a poskytnutí podkladů na kterých jsem stavěl.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 30. 5. 2007

Michal Bečka

# Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
<b>2</b>	<b>Obecné pojmy</b>	<b>6</b>
2.1	Počítačová síť . . . . .	6
2.2	Směrovací algoritmus . . . . .	7
2.3	Dlouhá zpráva . . . . .	7
<b>3</b>	<b>Použité techniky</b>	<b>8</b>
3.1	Směrovací algoritmus . . . . .	8
3.2	Přenos zprávy . . . . .	8
3.3	Saturace sítě . . . . .	9
<b>4</b>	<b>Návrh uzlu sítě</b>	<b>10</b>
4.1	Struktura . . . . .	10
4.2	Časování . . . . .	13
<b>5</b>	<b>Simulační program</b>	<b>14</b>
5.1	Vstupní data . . . . .	14
5.2	Výstupní data . . . . .	15
<b>6</b>	<b>Měření</b>	<b>16</b>
6.1	Přerušovací signál . . . . .	16
6.2	Závislost parametrů . . . . .	17
6.3	Doba mezi spojeními . . . . .	19
6.4	Ostatní typy sítí . . . . .	21
<b>7</b>	<b>Závěr</b>	<b>23</b>
	<b>Literatura</b>	<b>24</b>
	<b>Obsah přiloženého CD</b>	<b>24</b>
	<b>Tabulky</b>	<b>25</b>

Název práce: Simulace systému pro směrování dlouhých zpráv

Autor: Michal Bečka

Katedra: Aplikované matematiky

Vedoucí bakalářské práce: Prof. RNDr. Luděk Kučera, DrSc., Katedra aplikované matematiky Matematicko-fyzikální fakulty Univerzity Karlovy v Praze, Malostranské nám. 25, Praha 1

E-mail vedoucího: ludek@kam.mff.cuni.cz

Abstrakt: Tato práce se zabývá specifickým typem síťového směrování, vytvořeným pro řízení dlouhých zpráv. Cílem je zjištění optimálního nastavení tohoto směrování pomocí simulace. Součástí práce je návrh uzlu sítě pro směrování dlouhých zpráv, v jazyce VHDL, jež je následně simulován pro zjištění jeho výkonových vlastností. Další částí je simulační program pro otestování efektivity různých nastavení tohoto směrování. Na síti vytvořené z navrženého uzlu je pomocí tohoto simulačního programu naměřeno optimální nastavení, zejména co se týče propustnosti a doby přenosu zprávy. Toto nastavení je následně implementováno do simulačního programu.

Klíčová slova: směrování, dlouhé zprávy, simulace

Title: Simulation of long message routing

Author: Michal Bečka

Department: Department of Applied Mathematics

Supervisor: Prof. RNDr. Luděk Kučera, DrSc., Katedra aplikované matematiky Matematicko-fyzikální fakulty Univerzity Karlovy v Praze, Malostranské nám. 25, Praha 1

Supervisor's e-mail address: ludek@kam.mff.cuni.cz

Abstract: This work deals with specific type of routing, created for directing long messages. The goal is to determine optimal settings for this routing by simulation. Part of the work is a design of a network node of a network for routing long messages, in VHDL language, this design is simulated for determining its performance. Another path is a simulation program for performance testing of various settings of this routing. By this program, optimal settings are determined for a network composed of the designed nodes, primarily for latency and network throughput. These settings are then implemented in the simulation program.

Keywords: routing, long message, simulation

# Kapitola 1

## Úvod

Jako každý provoz musí být řízen, např. na křižovatce řídí provoz semaforey, dispečerů se starají o navigaci vlaků a letadel, tak i provoz v počítačových sítích nemůže být chaotický. Tam se o jednu z částí řízení provozu starají směrovací algoritmy, navigující datové zprávy přes uzly sítě od počátku k jejich cíli. Cílem této práce je otestovat jeden z takových způsobů směřování, zaměřeného na dlouhé zprávy. Právě kvůli jejich velké délce tento algoritmus využívá postupy, které by u normálních kratších zpráv přinášely komplikace a naopak se snaží vyvarovat postupům efektivně fungujícím na krátkých zprávách, ale nevhodných pro dlouhé zprávy., jako např. čekání v uzlu na uvolnění nejkratší cesty k cíli.

Samotný algoritmus je popsán ve Wait-Free Deflection Routing of Long Messages [1], na čemž tato práce staví a snaží se vytvořit jednoduchý prostředek k jeho otestování. Od návrhu základního kamenu sítě – uzlu, po kompletní simulační nástroj pro takovou síť, umožňuje práci nahlédnout do chování síťového provozu dlouhých zpráv, umožňuje hledání nastavení směrovacího algoritmu, které poskytne ty nejlepší výsledky na dané síti.

Tento směrovací algoritmus rozhodně není vhodný pro každou síť, většinou sítě jak je známe (internet, LAN) procházejí krátké zprávy, ale pro specifické sítě s malou délkou obvodů tento algoritmus určitě stojí za zamyšlení. Pro síťové návrháře by výsledky této práce mohly poskytnout dostatek podkladů k rozhodnutí, jestli je právě tento směrovací algoritmus to pravé pro jejich síť, či jestli jeho výkon nedostačuje jejich požadavkům.

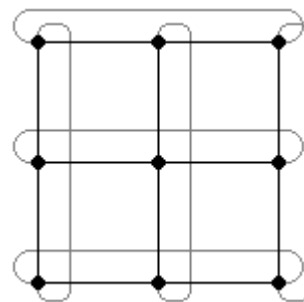
# Kapitola 2

## Obecné pojmy

### 2.1 Počítačová síť

Pro potřeby této práce definujeme počítačovou síť, je to souvislý graf, skládající se z uzlů (vrcholů grafu) a drátů (hran grafu), které uzly spojují. První použitý typ sítě je neorientovaný dvoj až čtyřrozměrný torus. Pro vysvětlení toru použitého v této práci – rozdíl mezi dvojrozměrným torem a čtvercovou sítí je ve spojení jeho okrajů, uzel na jednom okraji je spojen drátem s uzlem na opačné straně (Obr. 1). Takovýto způsob získává zajímavé vlastnosti, např. když postupujete jedním směrem od uzlu přes  $n$  hran/drátů, kde  $n$  je délka původní čtvercové sítě, vždy se dostanete zpět do toho uzlu. Další použitý typ grafu je De Bruijnův graf, orientovaný graf s unikátními vlastnosti pro směrování.

Všechny dráty mají stejnou délku, i ty spojující krajní uzly čtvercové sítě v toru, obrázek proto není co se týče délky drátů přesný. Díky tomu je zaručena jedna důležitá vlastnost – všechny uzly mají stejné vlastnosti, to platí pro oba typy použitých sítí. Neexistuje krajní uzel s nejvzdálenějším vrcholem dvakrát dále než u uzlu uprostřed sítě.



Obr. 1: 2D torus

Sítí prochází signály a zprávy, popsané níže v kapitole 2.3. Provoz v síti limituje fakt, že jedním drátem nemůže procházet více než jedna zpráva, jen jedním směrem. Zato uzel tento limit nemá, může spojovat komunikaci mezi několika dvojicemi drátů. Signály, stejně jako zprávy, mohou procházet v jednom drátu jen po jednom, avšak mohou procházet oběma směry zároveň. Také jsou nezávislé na zprávách, jedním drátem tedy teoreticky může najednou procházet jedna zpráva a dva signály různým směrem. To proto, že drát se skládá z datového drátu pro zprávy a dvou signálních drátů pro oba směry. U De Bruijnova grafu, přestože je orientovaný, jsou signální

dráty také dva, ale datovým drátem může zpráva procházet jen jedním směrem. Detailní vysvětlení počítačových sítí poskytuje kniha v referenci [2].

## 2.2 Směrovací algoritmus

V síti probíhají zprávy, každá zpráva je do sítě vložena přes nějaký uzel a má daný cílový uzel, do kterého má být přenesena. Z počátečního uzlu je vyslána po drátu do sousedního uzlu, který ji pak přepošle přes další drát k dalšímu uzlu, atd., dokud se zpráva nedostane do svého cílového uzlu. Aby zpráva jen tak nebloudila sítí, je zapotřebí v každém uzlu mít mechanismus, který vyhodnotí kam se zpráva chce dostat a pošle ji správným směrem. Tímto mechanismem je právě směrovací algoritmus. Zpráva je tedy znovu směrována v každém uzlu, kterým projde. Hlavním cílem je dostat zprávu co nejdříve ke svému cíli, což se většinou řeší snahou nalézt co nejkratší cestu. Ovšem to je komplikováno i obsazeností sítě, nejkratší cesta k cíli může být obsazená přenosem jiné zprávy.

Různým stylem směrování je možno dosáhnout značně rozdílných vlastností, hlavní vlastnost je latence – doba, za kterou je zpráva doručena, další vlastnosti se do ní už jen promítají.

## 2.3 Dlouhá zpráva

Zpráva v síti má svou délkovou míru – jestliže je celá na cestě, je to délka drátu kterou zaujímá, od začátku zprávy po její poslední bit. Dlouhá zpráva je přibližně tak dlouhá, že její délka přesahuje, někdy i mnohonásobně, průměrnou délku cesty v síti. V běžných simulacích je zpráva (paket) brána jako atomický bod, pohybující se drátem, což je zjednodušení, které si u délky dlouhé zprávy nemůžeme dovolit. Přesnější definice dlouhé zprávy je popsána v referenci [1].

Zpráva neprochází sítí rovnou, její přenos je řízen signály. Ty se nejdřív postarají o navázání spojení – rezervaci volné cesty od počátku k cíli, pak o její doručení a následně zase uvolnění cesty. Tedy směrování vybírá směr přenosu jen pro signál navazující spojení, pro ostatní signály a zprávu už si pamatuje vybraný směr a posílá je tam.

# Kapitola 3

## Použité techniky

Tato práce je praktickým ověřením směrovacího algoritmu popsaným v Wait-Free Deflection Routing of Long Messages [1], v této kapitole se pokusím shrnout a přiblížit to, co je tam o tomto algoritmu napsáno.

### 3.1 Směrovací algoritmus

Pro snížení latence se tento algoritmus snaží za každou cenu vyhnout čekání v uzlech během hledání cesty pro přenos zprávy, odklání komunikaci od obsazených drátů. Pokud ji není možné poslat směrem nejkratší možné cesty, vybere jiný směr, vedoucí na delší cestu, avšak s možností pokračovat ihned. Při malé délce zprávy by to nemusel být výhodný přístup, ne však při dlouhých zprávách. Doba čekání na uvolnění obsazeného drátu je totiž průměrně polovina doby, co je drát obsazen, což u dlouhých zpráv může být daleko delší doba, než o kolik se zpozdí přenos zprávy posláním odkloněnou delší cestou. Navíc v této práci je přenos zprávy spojovaný, zpráva se přenáší až po rezervaci přenosové cesty, což je optimální pro zprávy dokonce delší než celá cesta, v tom případě je výhodnost čekání na uvolnění drátu výrazně nižší než zvolení jakéhokoliv ihned přístupného směru.

### 3.2 Přenos zprávy

Přenos zprávy začíná posláním signálu na navázání spojení, pro nalezení cesty k cíli komunikace. Tento signál rezervuje pro svoji zprávu dráty, přes které přejde. Až dorazí do cíle, další signál je poslán zpět po cestě, aby počátku potvrdil úspěšné navázání spojení. Nyní, když je cesta vytvořena, počáteční uzel začne posílat zprávu. Až je zpráva doručena v cílovém uzlu, zpět po cestě je poslán další signál potvrzující přenos zprávy, zároveň uvolňuje zarezervované dráty při své cestě. Až dorazí na počátek, úspěšně ukončuje spojení.

To by byl ideální pokus o spojení, nyní omezující faktory. Po vyslání signálu pro vytvoření cesty, počáteční uzel naplánuje poslání přerušujícího signálu. Až přijde čas,



je poslán za prvním signálem, zároveň je započat přenos zprávy, aby se neztrácel čas. Jestliže přerušovací signál dostihne ten první než dorazí do cíle, je pokus o spojení ukončen a signál na uvolnění drátů je poslán zpět po dosud vytvořené cestě, čímž ukončí pokus o spojení neúspěchem. Pokud ale je cesta úspěšně vytvořena předtím, než přerušovací signál stihne pokus ukončit a ten se tedy potká s potvrzovacím signálem, je ten přerušující signál ignorován, a proto, že je následován zprávou, ta může pokračovat po cestě k cíli. Tím pak pokračuje úspěšný pokus o přenos zprávy. Přerušující signál zabraňuje zprávě hledat cestu příliš dlouhou, aby zbytečně nezabírala síťové prostředky.

Při přenosu dat zprávy je možné, že se data nedoručí správně, když cílový uzel zjistí takovou chybu, pošle uvolňovací signál a pokus spojení je ukončeno neúspěchem.

Když je zpráva poslána do sítě, je vložena do fronty svého počátečního uzlu, která má omezenou kapacitu. Jestliže fronta není prázdná a uzel se o žádnou komunikaci nepokouší, vyjme zprávu z fronty a pokouší se o její doručení. Uzel se nemůže pokoušet o více než jedno připojení najednou, vždy se zabývá maximálně jednou zprávou, k další se dostane až je ta před ní doručena. Když je fronta plná a další zpráva je přidána, tak se nezařadí a zahodí se.

### 3.3 Saturace sítě

Když se zaplnění sítě zvyšuje, spojení jsou delší, protože se musí vyhýbat. Může se stát, že se signál na tvoření cesty zasekne v jedné oblasti a zatímco se snaží najít cestu ven kličkováním z místa na místo, zaplní tuto oblast, čímž spustí dominový efekt, ostatní zprávy se také nebudou schopny přes tuto oblast dostat, což pak zaplní celou síť. Tento stav se nazývá “deflection jam”, saturace sítě, něco jako dopravní zácpa, čemu se snažíme vyhnout a zotavit se z toho co nejdříve. V běžných směrovacích algoritmech se neobjevuje, je to jedna špatná stránka tohoto směrovacího algoritmu, se kterou je potřeba se vypořádat.

Na to je právě určena doba mezi spojeními (v programu “resend time”). Jejím zvyšováním po neúspěšném pokusu o spojení se může počet nových spojení v síti snížit téměř na nulu, čímž dává možnost síti se uvolnit.

# Kapitola 4

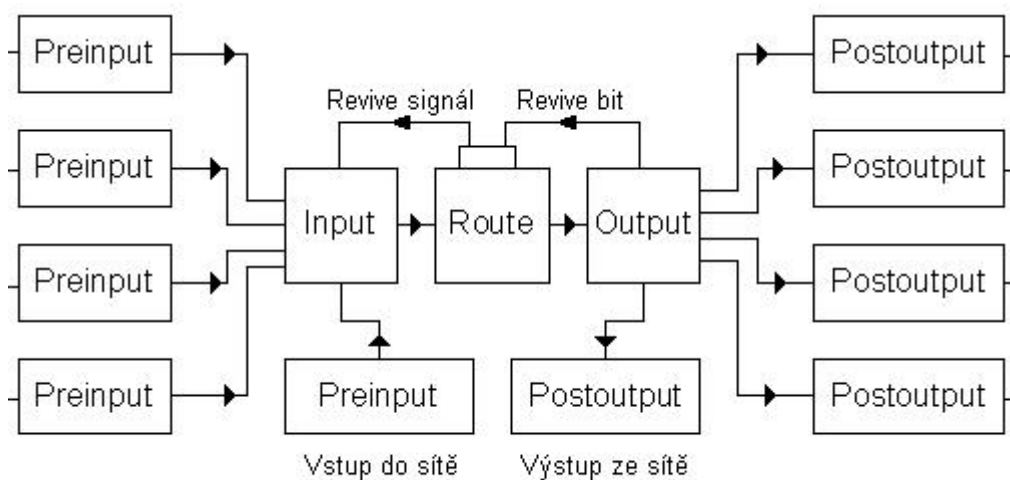
## Návrh uzlu sítě

Uzel je základním kamenem sítě, aktivní prvek. Dráty jsou jen pasivní části umožňující uzlům komunikaci. Pro potřeby simulace je o uzlu potřeba vědět, jak dlouho mu trvá zpracování úkolů. Pomocí následujícího návrhu je přibližně zjištěna doba směřování signálů. Tato hodnota je pak v dalších měřeních v této práci použita.

### 4.1 Struktura

Následující návrh uzlu zdaleka není nejoptimálnější, slouží jen pro přibližné zjištění konstanty. Jinými návrhy je určitě možno dosáhnout lepších výsledků, našim cílem je však alespoň řádově určit dobu směřování v uzlu, koneckonců tato doba je jedním z parametrů simulace, není fixní.

Navržen je uzel pro 2D torus, který má čtyři směry komunikace, navíc další dva, jeden směrem ven ze sítě a druhý nazpět. Přestože je toto návrh jen jednoho typu sítě, ukáže se, že počet vstupů má malý vliv na dobu zpracování signálu, proto výsledky mohou být použity i pro ostatní typy sítí zmíněných výše.



Obr. 2: Struktura uzlu

Uzel se skládá z jednotlivých modulů, které se starají o různé části zpracování signálu. Každý modul přijímá hodinový signál, při každém tiku tohoto signálu provede jednu operaci, jedno zpracování svých vstupů na svoje výstupy. Modulovou strukturu můžete vidět na obr. 2.

Dráty sítě mezi uzly, jak signální, tak datové, jsou sériové kabely, po kterých informace proudí bit po bitu, proto při vstupu do uzlu je potřeba přicházející informaci nejdříve dát dohromady. Na to slouží vstupní modul preinput, pro každý přichozí signální drát je přítomna jedna instance tohoto modulu. Při tiku hodin přijme bit ze signálního drátu a uloží si ho do své paměti. Je to stavový modul, při zjištění přenosu signálu přechází do dalšího stavu s každým dalším přijatým bitem, pokaždé tento bit přiloží k dosud přijaté části signálu ve své paměti. V posledním stavu vydá na svůj výstup signál jako celek. Samotný signál se skládá ze tří bitů pro typ signálu a v případě, že je to typ na tvoření cesty, nese sebou ještě index cílového uzlu, zvolíme velikost tohoto indexu 16 bitů.

Všechny moduly preinput předávají své výsledky do modulu input, který vybere jeden ze signálů a předá ho ke směrování do modulu route. Celkově vybírá signál mezi čtyřmi směry ze sítě, směru přicházejícího do tohoto uzlu z vně sítě a ze speciálního signálu revive, který bude popsán později. Vybrat jeden signál je potřeba, protože modul route nedokáže směřovat více než jeden signál najednou. Až input zvolí jeden ze signálů, předá ho dál spolu s indexem směru, ze kterého byl přijat. Signály, co nebyly vybrány, zůstanou nastaveny ve vstupu modulu input a při dalším tiku hodin se vybírá i z nich. Takto v podstatě doba zpracování nezávisí na tom, z kolika vstupů je vybíráno.

Modul route má za úkol vybrat k signálu směr, kterým ho poslat dál. Obsahuje dvě hlavní datové struktury - směrovací tabulku a tabulku navázaných spojení. Pokud je směrován signál pro vytváření cesty, je podle indexu cíle cesty vybrán řádek ve směrovací tabulce. Každý řádek se vztahuje k určitému intervalu indexů cílů, vybrán je ten, do jehož intervalu cílový index zapadne. Řádek dále obsahuje pořadí směrů, seřazených podle výhodnosti, na které signál poslat. Výsledný směr je vybrán jako první volný směr z tohoto pořadí. Protože se tímto signálem cesta tvoří, je

do druhé datové struktury, tabulky navázaných spojení, zaznamenáno spojení směru, ze kterého signál přišel, se směrem, na který je signál posílán.

Všechny ostatní typy signálů mají cestovat už po vytvořených cestách, na ně se tedy směrovací tabulka nevztahuje. Směr, kterým je poslat je jednoduše přečten z tabulky navázaných spojení podle směru, ze kterého signál přišel. Pokud je typ signálu rušení cesty, je z této tabulky spojení vymazáno. Podle ní se také propojují datové dráty, změnou tabulky navázaných spojení se příslušné dráty propojí, či odpojí.

Z route je signál poslán do modulu output, který signál jednoduše pošle na ten výstup, který byl modulem route vybrán. Jeden z výstupů je i směr ven ze sítě přes tento uzel, route tento směr vybere, pokud je signál určený tomuto uzlu. Na modul output navazují pro každý výchozí směr moduly postoutput, dělají přesně obráceně to, co dělaly moduly preinput. Celý signál rozdělí na jednotlivé bity a postupně ho posílají po svém jediném sériovém výstupu.

Dříve byl zmíněn speciální signál revive, ten je součástí mechanismu pro čekání signálů na vytvoření cesty, když nemají kudy z uzlu uniknout, když jsou všechny cesty obsazené. Jestliže modul route při výběru směru ze směrovací tabulky pozná, že signál nemá kam poslat, uloží si index cílového uzlu do paměti, stačí mít místo na jeden index pro každý směr, z jednoho směru nemohou přijít dva signály tvořící cestu najednou. Příchozí směr označí jako speciálně obsazený, čekající. Nyní je potřeba zajistit, aby byl čekající signál znovu směrován, to nemá cenu dokud je uzel pořád obsazený. Proto modul output kontroluje odeslané signály a když odesílá signál na rušení cesty, znamená to, že jestli byl uzel obsazený, tímhle rušením cesty se nějaké směry uvolnily. To je ten pravý okamžik se pokusit směrovat čekající signály. Modul output tedy pošle jednobitový signál části modulu route, která jen zjistí, jestli v paměti nějaké čekající signály jsou. Jestli ano, jeden vybere a pošle již zmíněný speciální signál revive do modulu input, spolu s indexem směru, ze kterého čekající signál kdysi přišel. Až je tento speciální signál inputem vybrán, je dále směrován, jako by to byl normální signál tvořící cestu, index cíle se vyvolá z paměti v route.

Zdrojové kódy tohoto návrhu v jazyce VHDL jsou na přiloženém CD.

## 4.2 Časování

Nyní k hlavnímu problému. Jak již bylo zmíněno, moduly jsou řízené hodinovými signály. Je však jasné, že ne všechny moduly mohou běžet na stejný hodinový signál, modulu preinput trvá několik tiků dát dohromady signál, zatímco modulu route trvá jen jeden tik ho přeměřovat. Proto jsou do uzlu vedeny dva hodinové signály, jeden rychlejší pro bitové skládání, preinput a postoutput, druhý pomalejší pro ostatní vnitřní moduly, starající se o jeden celý signál. Testování návrhu a jeho syntéza byly provedeny pomocí softwaru [4] firmy Xilinx® na čipu Virtex™ 4, konkrétně model xc4vlx25-12-sf363.

Perioda prvního hodinového signálu vyšla 4.976ns, tedy frekvence 200.950MHz. Perioda druhého 11.276, frekvence 88.688MHz.

Nyní se podívejme na dobu zpracování jednoho signálu. Na typ signálu stačí tři bity, na index cílového uzlu, v případě, že se jedná o signál pro vytvoření cesty, jich použijeme 16. Při nečinnosti je drát nastaven na 0, při začátku komunikace je vyslána jedna 1 pro identifikaci přicházejícího signálu. Celkově tedy signál potřebuje 20 tiků prvního hodinového signálu v modulu preinput. Dále je předán přes 3 moduly - input, route a output napojené na druhý hodinový signál. Na vyslání dál potřebuje dalších 20 tiků v modulu postoutput. Celkově tedy 40 tiků prvního signálu a 3 druhého signálu. To dohromady dává 232,868ns. Při zpracování signálu co netvoří cestu, tedy nemusí sebou přenášet index cíle, stačí jen 8 tiků prvního signálu, tedy 73,636ns. Pokud však vezmeme v úvahu fakt, že tuto dobu používáme spolu s dobou přenosu informace po drátu, pak správné zpoždění signálu v uzlu je doba od začátku zachycení prvního bitu signálu po vyslání prvního bitu výsledného signálu, ne jeho celé vyslání. Proto jsou potřebné hodnoty 20 tiků prvního signálu a 3 druhého signálu, tedy 133.348ns, pro ostatní signály 53,732ns. Samozřejmě uzel nezpracovává tuto dobu jediný signál, uzel může teoreticky zpracovávat další signál každých 11.276ns, ale to je vedlejší.

Pro potřeby simulace budeme uvažovat horní hranici, tedy zaokrouhleně 133ns.

# Kapitola 5

## Simulační program

Pro zjištění chování síťového provozu je potřeba ho otestovat. Postavení sítě a jeho reálné testování není vhodné jako první krok, je lepší si danou síť nejdříve vyzkoušet virtuálně – pomocí simulace. To umožňuje další část této práce, simulační nástroj pro síťový provoz dlouhých zpráv, využívající směrovací algoritmus popsáný výše. Tento program je rozšířením programu vytvořeného v ročníkovém projektu [3], změny se týkají především zmenšení paměťových nároků a usnadnění organizace simulací. Tato kapitola je velice stručná, samotné popsání simulačního programu je příliš rozsáhlé a je umístěno v dokumentaci na příloženém CD, stejně tak zdrojové kódy a spustitelný soubor. Program je psáný v anglickém jazyce.

### 5.1 Vstupní data – nastavení simulace

V této kapitole jsou popsány základní nastavení, které nás budou zajímat.

**Net type:**

Typ sítě, která má být simulována, dvoj až čtyř dimensionální torus nebo ještě De Bruijnův graf

**Theoretical load:**

Teoretické zatížení sítě, je to přibližná hodnota očekávaného zatížení. Přesná hodnota nemůže být vypočítána, protože závisí i na vzdálenosti počátku a cíle cesty zprávy, která je náhodná.

**Size:**

Pro tory, udává jejich velikost v každé dimenzi, např. šířka, délka. Je to počet uzlů, ne délková míra.

**Dimension:**

Pro De Bruijnův graf, nastavení dimenze, rozměru sítě.

**Symbols:**

Pro De Bruijnův graf, nastavuje počet symbolů sítě.

**Calculation of resend time**

Přepočítání doby mezi dvěma spojeními, měla by se zvýšit po neúspěšném spojení a snížit po úspěšném. Pro oba případy jsou tu kvadratické funkce s proměnnou old time – poslední doba mezi spojeními daného uzlu, tak se může tato doba plynule měnit.

**Node processing delay**

Jak dlouho uzlu trvá zpracování signálu. Toto je hodnota měřená v kapitole 4.

**Node to node traveling delay**

Jak dlouho trvá signálu/zprávě přenos přes jeden drát. Všechny dráty jsou stejně dlouhé, tak je to jedna hodnota.

**Abort delay**

Doba, za kterou vyslat přerušovací signál.

**Message length**

Jak dlouho trvá zprávu odeslat.

## 5.2 Výstupní data – výsledky simulace

Nabízí se několik výstupů – grafický, potom je tu log zpráv ale důležité jsou údaje ve shrnutí – summary. Údaje, co nás budou zajímat jsou:

**Average latency**

Průměrná latence, doba, kterou trvá zprávu doručit, od vložení do sítě po uvolnění cesty po úspěšném přenosu. Toto je doba, co je potřeba minimalizovat, je to nejdůležitější výsledek simulace.

**Average load**

Průměrné zatížení sítě.

**Average resend time**

Průměrná doba mezi spojeními.

**Number of lost messages**

Počet zahozených zpráv v důsledku přetečení fronty zpráv.

# Kapitola 6

## Měření

Většina parametrů sítě je pevně daná, např. její typ a velikost, či délka zpráv v ní procházejících. Některé je ale třeba prostě zvolit, jsou to doba, za kterou vyslat přerušovací signál a doba mezi dvěma pokusy o spojení. Cílem simulací je tedy určit po pevné parametry optimální volitelné parametry, tak aby se minimalizovala hlavní požadovaná vlastnost – latence. Následující měření ukážou jak důležité jednotlivé nastavení jsou a jakých výsledků je s nimi možno dosáhnout.

Základní měření jsou provedena na síti typu 2D torus, s velikostí 32 x 32. Pro takto velkou síť je dostatečné simulovat přenos 20000 zpráv, každým uzlem průměrně projde téměř 20 zpráv, což umožňuje síťovému provozu se rozumně ustálit. Minimální doba mezi spojeními bude 600ns. Fronta zpráv je pro toto měření eliminována, nastavením její kapacity na 1, každý uzel se může tedy zabývat pouze jednou zprávou a ostatní odmítá. Tento způsob usnadňuje rozeznání výsledků, porovnáváním počtu odmítnutých zpráv se dá jednodušeji porovnávat obsazení sítě co se týče volnosti uzlů, a zároveň zpřesňuje latenci, její součástí už není doba strávená čekáním v uzlu, pouze doba, kterou ji trvalo doručit. Možnost ztráty dat při přenosu nebudeme uvažovat. Doba zpracování signálu v uzlu je podle kapitoly 4 133ns, dobu pro drát zvolíme např. 167.

Výsledky tohoto základního měření budou ověřeny pro ostatní typy sítí a případně zjištěny rozdíly.

Nejprve se zaměříme na ty nastavení, která se ukázala jako málo důležitá, popřípadě zbytečná.

### 6.1 Přerušovací signál

Při všech měřeních se zvýšení doby, za kterou se poslat přerušovací signál, projevilo negativně. Latence se postupně zvyšovala spolu se snižováním propustnosti sítě.



V této simulaci směřování každého signálu uzlem trvá stejně dlouho, aby tedy přerušovací signál dohonil ten, co rezervuje cestu, musí ten první se nejprve úplně zastavit v nějakém uzlu. Aby se tak stalo, z tohoto uzlu nemůže vést žádný volný drát, protože jinak by se po něm signál vydal. Toto směřování se za každou cenu snaží vyhnout čekání, negativní vliv zvyšování tohoto nastavení tedy potvrzuje, že čekání v uzlu na uvolnění cesty je opravdu nevýhodné. Je lepší takto čekající signál ihned zrušit, uvolnit cestu a pokusit se o nové navázání spojení. Pro zbytek měření je toto nastavení zvoleno jako součet doby zpracování signálu uzlem a doby přenosu signálu přes jeden drát, čili dohromady je přerušovací signál hned jeden krok za tím cestu-tvořícím. Není to výhoda jen pro neúspěšná spojení, ale vzhledem k tomu, že spolu s přerušujícím signálem je rovnou posílána samotná zpráva, tak se zbytečně neztrácí čas potvrzováním úspěšného spojení.

Je ještě jedna možnost, která by mohla v některých návrzích nastat – že přerušující signál je směřován rychleji, než ten, co tvoří cestu. To se také děje v uzlu, co byl navržen v kapitole 4, ale to je způsobeno jen tím, že tvořící signál sebou musí ještě přenášet index cílového uzlu. Tímto stylem by bylo možné naměřením tohoto zpoždění efektivně omezit maximální délku vytvářené cesty, spočítat si, za kolik kroků přerušovací signál dožene náskok toho prvního. Tento způsob určitě stojí za zamyšlení, ale tyto simulace se jím nezabývají.

Bylo by tedy výhodné přímo v uzlu zakázat čekání na uvolnění cesty a při absenci volného drátu signál rovnou zrušit a poslat zpět. Přerušovací signál by ovšem stejně byl prospěšný uvozováním přenosu těla zprávy, to nemůže jít hned s prvním signálem, musí jít po už rezervované cestě, protože oba směry sdílí jeden datový drát. Posláním dvou tvořících signálů po jednom drátu proti sobě se nic nestane, jdou po různých drátech, avšak kdyby sebou měli i tělo zprávy, posílané dvakrát do jednoho drátu, mohlo by dojít k fyzickému poškození drátu.

## 6.2 Závislost parametrů

Doba přenosu informace po drátu a doba zpoždění signálu v uzlu mají na sebe zajímavý vliv. Vzhledem k tomu, že signál projde stejným počtem drátů, jako uzlů

(až na 1 uzel, ale to se při velké délce cesty ztratí), pak na simulaci má vliv pouze součet těchto hodnot. Zvyšováním jedné na úkor druhé proto na simulaci nemá vliv. (Proto taky zvolená doba drátu 167, ať je součet hezké číslo).

Jediná další nastavení určující čas v simulaci je délka zprávy a doba simulace. Násobení předchozího součtu a těchto časových hodnot stejným číslem (nazveme ho  $a$ ) se výsledky simulace nezmění, hodnoty jako zatížení sítě a vůbec všechny hodnoty, mimo těch časových, zůstanou stejné. Výsledné časové hodnoty se změní opět na příslušný násobek čísla  $a$ . Je to jako zvolit jinou jednotku pro všechny časové parametry.

V dalších měřeních nebude brána v potaz délka zprávy a součet doby přenosu informace po drátu s dobou zpoždění signálu v uzlu. Místo toho bude hlavním parametrem simulací jiná hodnota, která z nich vychází a to je teoretická zátěž sítě. Bude použita proto, že je jednodušší porovnávat vlastnosti sítě vzhledem k jedné hodnotě místo ke třem. Kromě předchozích hodnot je vypočítaná ještě z typu a velikosti sítě, počtu zpráv a doby simulace. Aby bylo tohle zjednodušení možné, je potřeba ukázat, že při stejné teoretické zátěži v jedné sadě měření různé hodnoty těchto parametrů nemají na simulaci vliv.

- Sada měření se provádí pro jeden typ a velikost sítě, ty tedy vliv nemají.
- Počet zpráv je také konstantní pro sadu měření.
- Doba simulace je lineárně závislá na teoretické zátěži. Při nastavení zbytku parametrů popsaných v dalším bodě je tedy možno změnou doby simulace nastavit požadovanou zátěž. Při stejné teoretické zátěži se tedy nastavení zbytku parametrů promítne pouze do doby simulace. Navíc, protože všechny časové parametry mohou být násobeny stejným číslem bez změn chování simulace (viz výše), se násobení zbytků parametrů stejným číslem promítne do doby simulace bez změny chování simulace. Proto chování simulace závisí jen na vztahu zbytku nastavení mezi sebou.
- Zbývají délka zprávy a součet zpoždění signálu přes drát a uzel. Vztah těchto

hodnot ukazuje tabulka č. 1, viz sekce Tabulky na konci práce. Při zvyšování délky zprávy vzhledem ke zmíněnému součtu se chování simulace ustálí. Pro různé velikosti sítí je poměr ustálení různý, to proto, že teoretická zátěž se nepočítá přímo z toho součtu, ale z průměrné délky cesty, která z něj vychází. Tedy potřebujeme, aby délka zprávy byla dostatečně větší, než průměrná délka cesty v síti. Ale to je už zaručeno tím, že jsou směřovány dlouhé zprávy, které by takové být měly, můžeme si tedy toto zjednodušení dovolit.

## 6.3 Doba mezi spojeními

Jak nastavit zvyšování této doby během krize a její následné snižování, až krize pomine, je dalším cílem simulace. Přestože je přepočítán kvadratickými funkcemi, ukázalo se, že kvadratický prvek je příliš drastický na určování času měřeného v tak malé jednotce jako jsou nanosekundy. Použity budou tedy jen lineární rovnice.

Nejdříve bude měřen lineární člen rovnice doby po neúspěšném pokusu, označme si ho jako FL, podobně lineární člen rovnice doby po úspěšném pokusu označíme SL. Výsledky simulací pro různé zátěže jsou v tabulce č. 2. U simulací, které skončily nezotavitelnou saturací sítě je uvedeno „saturace“.

Pro nižší zátěže (do 0.15) je z tabulky vidět, že jak v latenci, tak v počtu odmítnutých zpráv vyšší hodnota FL nevede k výrazným změnám hodnot, jejich rozdíly jsou malé a má na ně vliv spíše náhodnost simulace než jiné vlastnosti provozu. U latence je to zřejmé, u počtu odmítnutých zpráv je potřeba si uvědomit, jaký díl je to z celkového počtu zpráv. U tohoto měření je to 20000, tedy hodnoty se pohybují mezi jedním a dvěma procenty. I tak ale nižší FL celkově vede k mírně lepším výsledkům.

Při vyšších zatíženích, od 0.15 do 0.25, už se výhodnost nižšího FL projevuje výrazněji, síť už pomalu dosahuje svého potenciálu a saturace sítě začíná být problém, ta však nastává téměř výhradně při nastavení FL na 1 a to od zátěže 0.23

výše, stačí zvýšit tuto hodnotu jen nepatrně a už to stačí na zamezení saturace, viz tabulka č. 3.

Co se týče zátěží vyšších než 0.30 už se vyplatí lineární člen zvyšovat, podle tabulek č. 2 a 3 se optimální hodnota FL pohybuje mezi 1.2 a 1.5, ale taková zátěž už vede k velkému množství odmítnutých zpráv, od 25% při zátěži 0.3 po více než polovinu u zátěží nad 0.5. Síť už takovou zátěž přenést nestihne, proto je lepší se soustředit na zátěže, co síť zvládne a při nich je FL výhodné co nejmenší, ale dost velké na zabránění saturace, tedy 1.1. Bylo by možné hledat přesnější hodnotu mezi 1 a 1.1, ale to by vyžadovalo daleko delší simulace, a podle tabulek je zřejmé, že výkon z takové hodnoty by nemohl být o moc lepší, než ten, který je dosažen s hodnotou 1.1.

Nyní absolutní člen rovnice doby po neúspěšném pokusu. Z tabulky č. 4 je vidět, že jak pro vysokou, tak pro nízkou zátěž má vyšší nastavení absolutního členu negativní vliv v každé situaci, proto je ponechán na 0.

Dostáváme se k rovnici pro dobu po úspěšném pokusu o spojení. Tabulka č. 5 ukazuje, že jakékoliv nenulové nastavení SL má také negativní vliv. Samotné nastavení nenulové doby po úspěšném pokusu jen zvyšuje průměrnou dobu mezi spojeními, což se v minulých měřeních ukázalo jako nevýhodné. Lepší bylo zachovávat ji jen tak vysokou, aby zabránila saturaci, ne víc, což se v těchto měřeních potvrdilo. Po úspěšném spojení se posílá nová zpráva úplně jiným směrem, proto zřejmě nemá cenu čekat kvůli událostem, které se mohou dít v jiné části sítě.

## 6.4 Ostatní typy sítí

Nyní otestujeme, jak se vztahují předchozí výsledky na jiné velikosti a typy sítí. U ostatních torů budou v tabulkách uvedeny výsledky měření s různým FL pro malé zátěže 0.10 a 0.15 a ještě pro jednu zátěž a to pro tu, při které se poprvé objeví saturace sítě při FL=1.

První budou 2D tory, velikosti 48 x 48 a 64 x 64. Na rozumné ustálení je potřeba simulovat alespoň 40000 zpráv, téměř 10 zpráv na uzel u 64 x 64, sice je to menší počet než u minulých měření, ale grafy ukázaly, že 20 zpráv na uzel je pro ustálení provozu mnohem víc než dostatečné. Výsledky jsou v tabulkách č. 6 a 7, potvrzují vlastnosti naměřené na velikosti 32 x 32, pro nižší zátěže má FL malý vliv a když začne hrozit saturace, je lepší nastavené mírně nad 1, tedy kolem 1.2.

Následuje 3D torus, velikosti 16 x 16 x 16 a 24 x 24 x 24. Na ustálení se spokojíme s 60000 zprávami, 5 na uzel. Výsledky udávají tabulky č. 8 a 9. Opět stejné vlastnosti, jen FL je vhodné 1.2. U toru 16 x 16 x 16 při zátěži 0.24 je možné pozorovat mírné zlepšení latence se zvýšením FL, avšak toto zlepšení je zastíněno snížením propustnosti.

Aby se 4D torus rozumně zaplnil, muselo by se otestovat obrovské množství zpráv, proto zvolíme menší velikost 8 x 8 x 8 x 8 (relativně menší, počet uzlů je řádově stejný), kde stačí simulovat 40000 zpráv pro 10 zpráv na uzel. Snížíme i velikost zprávy, dost dlouhá stačí i 15000ns. Výsledky ukazuje tabulka č. 10. Vlastnosti se už se změnou FL nemění plynule, při malých zátěžích je jako u ostatních malý vliv očekávaný, ovšem při větším zatížení se zdá jedno, kterou hodnotu má FL v rozmezí 1.1 a 1.5. Na jedné straně pod 1.1 nastává saturace, na druhé straně nad 1.5, kolem 2 se vlastnosti také rapidně zhorší.

Dostáváme se k De Bruijnově grafu, jeho vlastnosti se ukázaly jako velice rozdílné od torů, především v tom, jak vypadá saturace sítě. Provoz má skvělé výsledky, dokud nenastane saturace, šance na navázání spojení je téměř stoprocentní. Pokud ale saturace nastane, je velice obtížné se z ní dostat. Její nástup je daleko snadněji

rozlišitelný než u torů, chování sítě se drasticky zhorší, tento skok však není ani tak způsoben horší saturací, jako lepšími výsledky, když síť saturovaná není. Navíc se saturace objeví náhle po dlouhé době naprosto bezproblémové simulace, tento jev byl u torů tlumen přerušovacím signálem, u tohoto typu sítě přerušovací signál stačí jen na udržení saturace před úplným zablokováním sítě.

Pro simulaci základních vlastností zvolíme počet symbolů 6 a dimenzi 4. Měření provádíme jen pro různé lineární členy obou rovnic, absolutní členy nepoužíváme, protože ty slouží na rychlejší zvýšení doby při malých hodnotách a k pomalejšímu zvyšování při velkých hodnotách, je potřeba přesný opak. Kvadratické členy se opět ukázaly jako příliš drastické, saturaci sice vyřeší, ale zablokují uzly obrovskou dobou mezi spojeními téměř napořád, po pár saturacích pak prázdná síť jen čeká. Proto tedy měření jen lineárních členů. Je zajímavé, že se rovnice uplatní pouze při saturaci, na normální provoz nemají vliv, protože šance na spojení je téměř stoprocentní.

Vyhodnocení výsledků není možné jen porovnáním statistických výsledků, někdy saturace nastane, někdy ne, proto jsou daleko lepším ukazatelem grafické výstupy simulace. Z nich vyplynulo, že zvyšováním FL se při saturaci odsouvá zátěž sítě na větší dobu po zotavení ze saturace. Zvyšováním SL k jedné zase napomáhá rychlejší reakci na saturaci. Nejmenší FL, co zvládne uvolnit síť, se pohybuje kolem 6, vyšší hodnota už jen zbytečně brzdí provoz, optimální SL se ukázalo jako 2/3.

# Kapitola 7

## Závěr

Směrování dlouhých zpráv má jiné vlastnosti, než v sítích běžně vidíme. Pro jeden směrovací algoritmus jsme ukázali, jak takový provoz dlouhých zpráv probíhá a jak může síť pro tento provoz vypadat. Otestovali jsme různá nastavení směrovacího algoritmu, zhodnotili jejich efektivitu a zjistili optimální nastavení. Některá nastavení se ukázala jako zbytečná, ale mezi nimi se objevilo i to, které svým použitím provozu v síti výrazně prospělo – lineární zvyšování doby mezi dvěma pokusy o spojení, jímž se většinou podařilo vyhnout se hlavnímu problému - saturaci sítě.

Naměřené výsledky umožňují udělat si obrázek o potenciálu tohoto směrovacího algoritmu, simulační program poskytuje možnost zjistit jeho výkon přímo na míru, což může pomoci síťovým návrhářům v rozhodování mezi tímto a jinými směrovacími algoritmy.

# Literatura

[1] Luděk Kučera (2001): Wait-Free Deflection Routing of Long Messages. IEEE Transactions on Parallel and Distributed Systems-May 2001 (Vol. 12, No. 5): 476-488

[2] F.T. Leighton (1991): Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes. Morgan Kaufmann Publishers.

Základ simulačního programu:

[3] Michal Bečka: Ročníkový projekt Simulace systému pro směrování dlouhých zpráv (2006/2007)

Použité internetové stránky:

<http://www.vhdl-online.de/tutorial/>

<http://www.xilinx.com/>

Použité vývojové nástroje:

Microsoft® Visual Studio® 2005

[4] ISE™ WebPACK™ - Xilinx® Integrated Software Environment (ISE) 8.1i

## Obsah přiloženého CD:

Elektronická forma této práce ve formátu PDF

Simulační program – zdrojové kódy v C++, dokumentace, projekt pro Microsoft® Visual Studio® 2005

Návrh uzlu – zdrojové kódy ve VHDL, projekt pro ISE™ WebPACK™



# Tabulky

Značení:

zátěž: teoretická zátěž

FL: lineární člen rovnice doby mezi spojeními po neúspěšném pokusu

FA: absolutní člen rovnice doby mezi spojeními po neúspěšném pokusu

SL: lineární člen rovnice doby mezi spojeními po úspěšném pokusu

Naměřené výsledky jsou prezentovány většinou ve dvojici, kde první číslo znamená počet zpráv odmítnutých sítě z důvodu zaneprázdněnosti uzlu – tato hodnota odráží propustnost sítě a druhé číslo je průměrná latence v nanosekundách

Tabulka č. 1:

Měřeno na 2D toru 32x32 pro zátěž 0.2, FL = 1.1.

poměr délky zprávy k součtu zpoždění signálu v uzlu a drátu

	40	100	200	400	800	1600
ztrac.zpráv	4022	797	671	633	587	583

Latence v této tabulce není směrodatná

Tabulka č. 2:

Měřeno na 2D toru 32x32.

FL

zátěž	1		1.5		2	
	ztrac.zpráv	latence(ns)	ztrac.zpráv	latence(ns)	ztrac.zpráv	latence(ns)
0.05	91	61941.1	104	61918.5	105	62048.6
0.10	221	65507.7	249	65859.8	255	65589.4
0.15	427	73725.7	394	75470.9	425	77595.6
0.17	499	79706	518	81688.3	552	87213.2
0.20	698	97397.7	777	100692	1030	130203
0.23	saturace		1626	169926	2117	194773
0.25			2490	223259	3152	254171
0.30			4810	323373	5268	320105

zátěž	1		1.5		2	
	ztrac.zprávk	latence(ns)	ztrac.zprávk	latence(ns)	ztrac.zprávk	latence(ns)
0.35	saturace		6672	386901	6944	433327
0.40			8161	414648	8307	432730
0.50			10268	450282	10287	411329

Tabulka č. 3:

Měřeno na 2D toru 32x32.

FL

zátěž	1		1.1		1.2	
	ztrac.zprávk	latence(ns)	ztrac.zprávk	latence(ns)	ztrac.zprávk	latence(ns)
0.10	229	65331.5	253	65203.6	277	65154.9
0.21	839	105910	807	109308	878	108085
0.30	saturace		5002	371392	4820	348280
0.40			8710	494721	8239	454043
0.60			12032	560518	11733	493412
0.80			13813	588631	13513	533063

Tabulka č. 4:

Měřeno na 2D toru 32x32.

FA

zátěž	FL		500	1000	2000	4000
0.20	1.1	ztrac.zprávk	709	733	744	783
		latence(ns)	97126.1	99079.9	100961	103697
	2	ztrac.zprávk	1049	1006	1031	1025
		latence(ns)	123673	125825	136362	136116
0.80	1.1	ztrac.zprávk	13478	13529	13424	13401
		latence(ns)	634763	689930	729929	731543
	2	ztrac.zprávk	13416	13481	13408	13415
		latence(ns)	490041	487576	539778	520223

Tabulka č. 5:

Měřeno na 2D toru 32x32.

SL

zátěž	FL		1	1/2	1/4
0.20	1.2	ztrac.zpráv	969	728	719
		latence(ns)	125896	98209	96418.1
	1.5	ztrac.zpráv	1915	867	855
		latence(ns)	247070	110334	113251
	1.8	ztrac.zpráv	2556	1293	997
		latence(ns)	323319	153671	132320
0.80	1.2	ztrac.zpráv	13611	13520	13487
		latence(ns)	1180530	905935	751107
	1.5	ztrac.zpráv	13935	13419	13497
		latence(ns)	1224160	801312	677239
	1.8	ztrac.zpráv	14196	13544	13518
		latence(ns)	1219450	811983	659415

Tabulka č. 6:

Měřeno na 2D toru 48x48.

FL

zátěž		1.1	1.2	1.5	2
0.10	ztrac.zpráv	324	345	342	382
	latence(ns)	72690.1	72875.7	72609.8	73502
0.15	ztrac.zpráv	632	618	590	636
	latence(ns)	88993	89291.5	90527.6	93033.6
0.19	ztrac.zpráv	530	653	683	1161
	latence(ns)	147025	149446	151020	210379

Tabulka č. 7:

Měřeno na 2D toru 64x64.

		FL			
zátěž		1.1	1.2	1.5	2
0.10	ztrac.zpráv	243	289	275	268
	latence(ns)	80790.2	81288	81019.3	81931.4
0.15	ztrac.zpráv	508	528	512	556
	latence(ns)	107389	107984	108405	125609
0.18	ztrac.zpráv	3260	1321	1381	1897
	latence(ns)	473466	222589	218751	258562

Tabulka č. 8:

Měřeno na 3D toru 16x16x16.

		FL			
zátěž		1.1	1.2	1.5	2
0.10	ztrac.zpráv	1869	1919	1906	1955
	latence(ns)	59563.9	59501.7	59511.5	59484.9
0.15	ztrac.zpráv	2911	2939	2940	2899
	latence(ns)	61378.6	61704.3	61674.8	61641.7
0.24	ztrac.zpráv	9816	9688	9956	10427
	latence(ns)	172192	160438	152164	151504

Tabulka č. 9:

Měřeno na 3D toru 24x24x24.

		FL			
zátěž		1.1	1.2	1.5	2
0.10	ztrac.zpráv	983	925	916	894
	latence(ns)	63618.9	63645.6	63532.6	63652.1
0.15	ztrac.zpráv	1478	1448	1486	1552
	latence(ns)	66271.9	66463.1	66511.6	66349
0.18	ztrac.zpráv	2040	2029	2034	2121
	latence(ns)	76239.8	76569.2	77704.3	80237.8

Tabulka č. 10:

Měřeno na 4D toru 8x8x8x8, pro délku zprávy 15000ns.

		FL			
zátěž		1.1	1.2	1.5	2
0.10	ztrac.zpráv	1818	1826	1783	1812
	latence(ns)	21306.6	21325.4	21334	21339
0.15	ztrac.zpráv	2662	2722	2738	2642
	latence(ns)	21725.1	21743.5	21731.4	21717.2
0.20	ztrac.zpráv	3652	3622	3663	5700
	latence(ns)	23016.5	22756.9	22680.5	33262