



**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ**

**Факултет „Компютърни системи и технологии”**

**Катедра „Компютърни системи”**

**Маг. инж. Иле Димитриевски**

## **ПРОЕКТИРАНЕ НА СМАРТ ETHERNET СУИЧ**

### **А В Т О Р Е Ф Е Р А Т**

на дисертация за придобиване на образователна и научна степен

**"ДОКТОР"**

Област: 5. Технически науки

Професионално направление: 5.3 Комуникационна и компютърна техника

Научна специалност: „Компютърни системи, комплекси и мрежи”

**Научен ръководител: доц. д-р инж. Валентин Стоянов Моллов**

СОФИЯ, 2019 г.

Дисертационният труд е обсъден и насочен за защита от Катедрения съвет на катедра „Компютърни системи “ към Факултет ФКСТ на ТУ-София на редовно заседание, проведено на 07.01.2019 г..

Публичната защита на дисертационния труд ще се състои на 28.03.2019 г. от 13:00 часа в Конферентната зала на БИЦ на Технически университет – София на открито заседание на научното жури, определено със заповед № ОЖ- 5.3-01 / 15.01.2019 г. на Ректора на ТУ-София в състав:

1. доц. д-р Петко Стоянов – председател
2. доц. д-р. Валентин Моллов – научен секретар
3. Проф. д-р. Иван Куртев
4. доц. д-р Стефан Димитров
5. доц. д-р Карло Чула

Рецензенти:

1. доц. д-р инж. Петко Стоянов
2. доц. д-р Стефан Димитров

Материалите по защитата са на разположение на интересуващите се в канцеларията на ФКСТ, каб. 1443А на ТУ – София.

Дисертантът е редовен докторант към катедра „Компютърни системи“ на Факултета по Компютърни Системи и Технологии (ФКСТ). Изследванията по дисертационната разработка са направени от автора, като някои от тях са подкрепени от научноизследователски проекти.

Автор: маг. инж. Иле Димитриевски

Заглавие: Проектиране на смарт Ethernet суич

Тираж: 30 броя

Отпечатано в ИПК на Технически университет – София

# **I. ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД**

---

## **Актуалност на проблема**

Големия обем на данни, който трябва да се предоставят и да се обменят във всички области където е необходимо започвайки с технологиите за разпространение на цифрови телевизионни и интернет услуги както за всекидневни нужди така и за обикновени потребители използва широколентови комуникационни мрежи най-често с оптическа среда за пренасяне на сигнала. Не всякога средата за пренасяне на цифровите данни е необходимо да е само оптическа, понякога се прилага и среда за пренасяне с меден проводник т.е. коаксиален кабел. Проектирането на високоскоростен комутатор (суич) за бъдещи поколения локални и градски Ethernet мрежи на база на тези модерни Мрежи на Чип (МнЧ) прави проблема дисертабилен.

В самата база на Science Direct се откриват 5467 статии, с ключови думи „network on chip” или МнЧ.

## **Цел на дисертационния труд, основни задачи и методи за изследване**

Цел на дисертационния труд е проектиране на високоскоростен комутатор (суич) за бъдещи поколения локални Ethernet мрежи с възможност за осигуряване на високо качество на услугите (QoS) чрез използване на Мрежи на Чип.

При избиране на основните задачи които са поставени по дисертационния труд са разделени на две части и това:

1. Избиране на оптимална архитектура по отношение на:
  - Малка стойност на латентността;
  - Голяма стойност на пропускателната способност.
2. Избиране на подходящ маршрутизиращ алгоритъм по отношение на:
  - Малка стойност на латентността;
  - Устойчивост на мъртво блокиране;
  - Устойчивост на живо блокиране;
  - Изразходована енергия.

## **Научна новост**

Тези нови типове на МнЧ може да се използват при високоскоростни модерни типове на локални Ethernet мрежи, които са с широка пропускателна способност, и се отличават с малка стойност на латентността. Комуникацията между два сървъра се нуждае от такава скорост и малка стойност на латентността. Но тук не свършва тяхното приложение - те могат да се използват и при интегриране на специални блокове на тяхната пластина от силиций с цел постигане на висока скорост на обработка на данните. Пример за това

приложение може да са модерни аудио-визуални контролери за кодиране/декодиране с широка пропускателна способност, които да са с ускорена обработка.

### **Практическа приложимост**

Практическата приложимост може да се види при високоскоростни комуникации - между сървър-сървър комуникация или пък в новите клетъчни GSM комуникации за комуникация помежду сървър-бекхаул (backhaul) т.е за разпръскване на модерни типове на протоколи било това да са 3G, 4G и 5G стандарт.

### **Апробация**

Научноизследователски проект в помощ на докторанта бр. 161ЦФ002-09 на тема „Изграждане на тест - платформа за оценка на качествените параметри на NoC –(Network on Chip)” Ръководител: доц. д-р инж. Валентин Стоянов Моллов

### **Публикации**

Основни постижения и резултати от дисертационния труд са публикувани както следва: 2–в рецензирани научни списания в страната, 1–в сборник на международна научна конференция и 4–в сборници от международни конференции в страната с международно участие.

### **Структура и обем на дисертационния труд**

Дисертационният труд е в обем от 182 страници, като включва увод, 4 глави за решаване на формулираните основни задачи, списък на основните приноси, списък на публикациите по дисертацията и използвана литература. Цитирани са общо 74 литературни източници, като 74 са на латиница, а останалите са интернет ресурси. Работата включва общо 117 фигури и 13 таблици. Номерата на фигурите и таблиците в автореферата съответстват на тези в дисертационния труд.

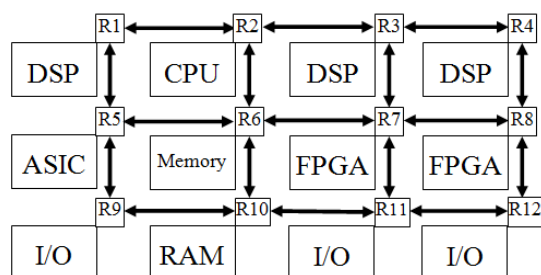
## **II. СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД**

---

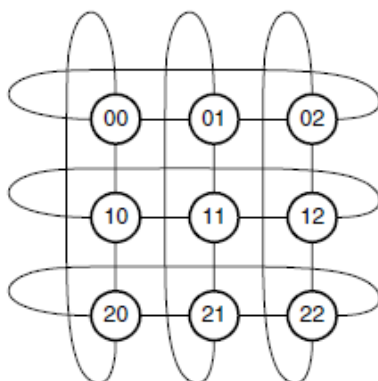
### **ГЛАВА 1. Обзор на инженерни решения. Научни проблеми**

Разглеждат се основните видове на мрежи които са от тип решетка фиг.1.1 (Mesh или Cliché) както и мрежа от тип торус фиг.1.2. В глава 1 имаме четири части, като в първата част е представен задълбочен анализ за това какво

представлява широчина на канала, неговата дължина, скоростта на предаване на данните през каналът за комуникация, стойността на латентността.



Фиг.1.1 Топология решетка (Mesh или Cliché) [9, 10]



Фиг.1.2 Топология 3 - масив 2 - куб (това означава 2D размерен торус)

От този обзор и инженерни решения, предлагани и използвани през годините, важно е да се получат количествени стойности за необходимите параметри за всяка от разгледаните видове на мрежи. Всичко това е подробно описано и е направен анализ чрез необходимите математически зависимости, и определяне на величините с които да стане ясно същността на стойностите на тези за нас важни величини, които ще се измерват и изследват. Във втора част на глава 1 [заглавие 1.6] са описани видовете маршрутизиращи алгоритми по отношение на тяхната главна класификация на групи по отношение на тяхното поведение при маршрутизация. Втората част от глава 1 е необходима, за да ни предостави информация за маршрутизиращите алгоритми, които ще оценяваме по отношение на малката стойност на латентността, устойчивост на мъртво блокиране, устойчивост на живо блокиране и изразходвана (потребявана) енергия. В глава 1 са дадени и описание на моделите на трафикът в таблица 1.1, които симулаторът [46, 6] ще използва при оценка на изследваните топологии [9, 10]. Тези модели на трафик се използват за оценяване на качествените показатели на топологията на мрежата, изброени по-горе. В третата част от глава 1 [заглавие 1.13] се разглежда част от материала, който се отнася за управление с потока на данните и дали това ще става естествено, или чрез самия алгоритъм на маршрутизация. В четвъртата част от глава 1 [заглавие 1.25] разгледаме характеристика на мъртво блокиране и живо блокиране. Докато в петата част от глава 1 [заглавие 1.30] ще се разгледат особеностите и съставните части при изграждане на маршрутизатори.

Таблица 1.1 - Модели на мрежов трафик

| Име на трафикът      | Уравнение                       |
|----------------------|---------------------------------|
| Случаен              | $\lambda_{sd} = 1/N$            |
| Пермутация           | $d = \pi(s)$                    |
| По битова пермутация | $d_i = s_{f(i)} \oplus g(i)$    |
| Бит допълнение       | $d_i = \neg s_i$                |
| Изменящ се бит       | $d_i = s_{b-i-1}$               |
| Ротация на бит       | $d_i = s_{i+1} \bmod b$         |
| Изместващ се бит     | $d_i = s_{i-1} \bmod b$         |
| Трансформираш се     | $d_i = s_{i+b/2} \bmod b$       |
| Пермутация на число  | $d_x = f(s_{g(x)})$             |
| Торнадо              | $d_x = s_x + ([k/2]-1) \bmod k$ |
| Съседно              | $d_x = s_x + 1 \bmod k$         |

$$\Theta_{идеално} = \frac{b}{\gamma_{\max}} \quad (1)$$

Потока на данни в този случай се определя с уравнение 1.

Докато пак при необходимост от най-високо предоставяне на данни трябва да разберем че попадането в напречните сечения на мрежата ще предизвика най-висока скорост за предаване на данни.

$$\gamma_{\max} \geq \gamma_B = \frac{N}{2B_C} \quad (2)$$

## 1.6 Алгоритми на маршрутизация

### 1.6.1 Детерминирани алгоритми на маршрутизация

Най-простите алгоритми за маршрутизация са детерминирани – те изпращат всеки пакет от източника  $x$  до местоназначение  $y$  през точно един и същ маршрут. Маршрутизиращата връзка за детерминиращия маршрутизиращ алгоритъм е дадена с функцията:

$$R: N \times N \rightarrow P \quad (12)$$

Докато в таблица 1.2 е даден параметърът за пропускателната способност при различни типове на маршрутизиращи алгоритми при различни топологии. Оценката е направена на базата на дискретния симулатор Booksim.

### 1.6.2 Пространствено маршрутизиране в куб-овите мрежи

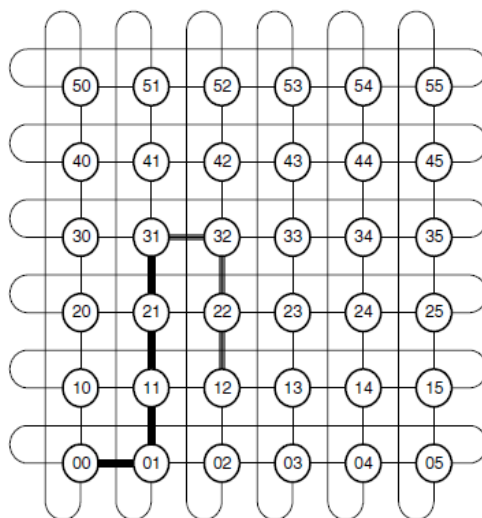
Пространственото маршрутизиране е аналог на дестинационно-етикетовото(тагово) маршрутизиране на директни  $k$ -масив  $n$ -куб мрежи (торус и решетка). [15]

Като пример за пространствено маршрутизиране, вземаме за пример пътуване на пакета от възел  $s = 03$  до възел  $d = 22$  в 6-масив 2-куб, показан на фигура 1.7. [15,42].

За изчисление на предпочитаната посока се използва уравнението:

$$\Delta_i = m_i - D_{T,i} = \begin{cases} 0 \text{ ако } |\Delta_i| \leq \frac{k}{2}, \\ \text{знак } (\Delta_i) \text{ в други случаи} \end{cases}, \quad (15)$$

където  $T$  показва че функцията е за торус топология.



Фиг.1.7 Пример за пространствено маршрутизиране източник 0,3 до местоназначение 22. В случая е приложено е XY маршрутиране

### 1.6.3 Несъзнаващи маршрутизиращи алгоритми

Несъзнаващо маршрутизиране е маршрутизирането на пакетите без да се вземе предвид състоянието на мрежата. Това е лесно да се изпълни и много просто да се анализира. Докато добавянето на информация за състоянието на мрежата може потенциално да подобри производителността на маршрутизацията, но също така добавя значителната сложност и ако не се изпълни внимателно, може да доведе до деградация на производителността. Основният компромис с несъзнаващо маршрутизиране е между местонахождането и управлението на трафикът.

След това, независимо от първоначалния модел на трафик, на всеки етап от алгоритъмът на Valiant изглежда че е еднороден случаен трафик [63]. По този начин алгоритъмът на Valiant намалява натоварването на всеки модел на трафик до два пъти от натоварването на случаен трафик, или на половината от натоварването на мрежата.

Алгоритъмът на Valiant дава добри резултати, в най-лошия случай при мрежите  $k$ -масив  $n$ -куб за сметка на местността. Всеки от двата случайни етапи

изпраща всеки пакет на средно разстояние от  $k / 4$  във всяка от  $n$  пространството за общ брой скокове върху двата етапи на  $nk / 2$ . Тогава всяка връзка има средно натоварване  $\gamma = k / 4$  и пропускателна способност от  $4b / k$ .

Тази пропускателна способност е почти оптимална, както се вижда от торнадо трафик модела.

При торнадо трафик модела, всеки пакет трябва да премине  $H = n (k / 2 - 1)$  скок (само при условие че  $k$  е четно число) [63].

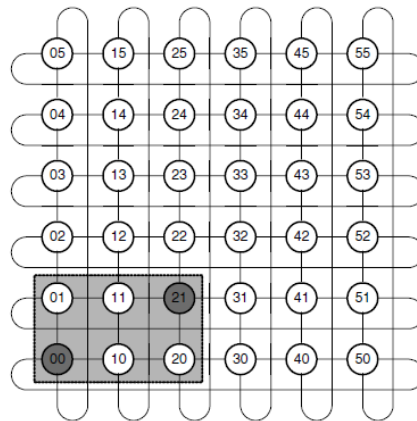
Натоварването на каналът за този модел при всеки алгоритъм на маршрутизация трябва да бъде поне колкото е дадено с уравнение 20.

$$\gamma \geq \frac{H_{min}N}{C} = \frac{n \left( \frac{k}{2} - 1 \right) N}{2nN} = \frac{k}{4} - \frac{1}{2} \quad (20)$$

### 1.6.5 Минималното несъзнателно маршрутизиране

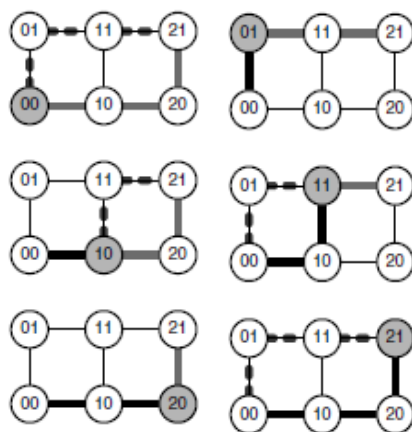
Минималното несъзнателно маршрутизиране се опитва да постигне управление на трафика при обхождащо маршрутизиране без да се отказваме от местността, но като ограничаваме маршрутите до минимални т.е най-кратък път.

Минимална версия на алгоритъма на Valiant може да бъде реализирана на  $k$ -масив  $n$ -куб топологии чрез ограничаване на междинния възел  $x$ , за да се намира в минималния квадрант между  $s$  и  $d$ . Минималният квадрант е с най-малка  $n$ -размерна подмрежа, която съдържа  $s$  и  $d$  като ъглови възли.



Фиг.1.9 Пример за Valliant маршрутизиране 6 - масив 2 - куб.

След като бъде идентифициран минималния квадрант, се избира междинен възел  $x$  от вътрешността на квадранта. След това пакетът се пренасочва от  $s$  към  $x$ , и след това от  $x$  до  $d$  чрез маршрутизиране на е-куб. В този случай има шест възможности за  $x$  и всяка от тях е илюстрирана от засенчените възли на фигура 1.10.



Фиг.1.10 Възможни стъпи на маршрутизиране при Valliant алгоритъм

### 1.6.6 Анализ на несъзнателно маршрутизиране

Тъй като несъзнателно маршрутизиращият алгоритъм избира път, независим от състоянието на мрежата, натоварването  $\gamma_c (sd)$ , предизвикано на канал с при единица трафик, изпращащ се от  $s$  до  $d$  (това е,  $\lambda_{sd} = 1$ ) е независим от трафикът, който се изпраща между всички други двойки възли. Тази особеност ни позволява да оценим трафика през канал за даден алгоритъм и за модел на трафика чрез сумиране на приноса на всеки елемент от матрица на трафика и е дадена чрез уравнение 21.

$$\gamma_c = \sum_{i,j} \lambda_{ij} \gamma_c(i,j) \quad (21)$$

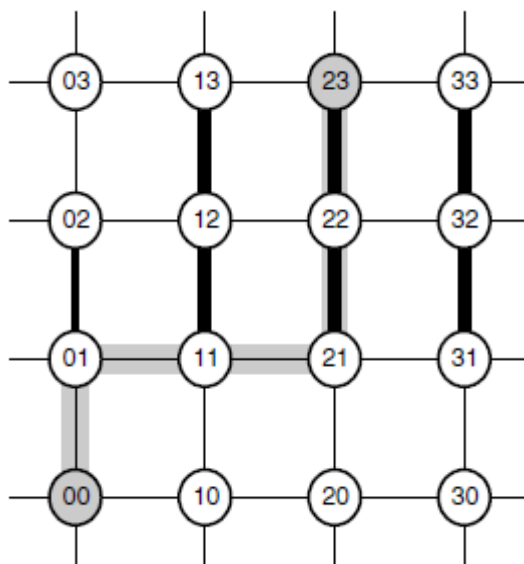
В таблица 1.2 е дадена пропускателната способност за четири несъзнателни алгоритми на маршрутизация, приложни при торус топология.

Таблица 1.2 Пропускателната способност както част от капацитета изследван при четири алгоритми на маршрутизация на 8-масив 2-куб при четири различни модели на трафик флитове. Както може да се забележи от таблицата, при модела на трафика при най-лошия случай е различен [9].

| Модел на трафикът | е-куб | Valiant | Минимален | Управление на трафика |
|-------------------|-------|---------|-----------|-----------------------|
| Най-близък съсед  | 4,00  | 0,50    | 4,00      | 2,33                  |
| Еднороден         | 1,00  | 0,50    | 1,00      | 0,76                  |
| Бит допълнение    | 0,50  | 0,50    | 0,40      | 0,2                   |
| Разместващ се     | 0,25  | 0,50    | 0,54      | 0,57                  |
| Торнадо           | 0,33  | 0,50    | 0,33      | 0,53                  |
| Най-лош случай    | 0,25  | 0,50    | 0,21      | 0,31                  |

В таблица 1.2 е дадена параметърът за пропускателната способност при различни типове на маршрутизиращи алгоритми при различни топологии.

Първоначалният скок е направен на север до възел 01. Във възел 01 връзката със север е леко претоварена - претоварването се обозначава със удебелена линията, така че следващият пакет се движи на изток до 11. След този скок на изток, всички пътеки на север са силно претоварени, така че пакетът е принуден да пресече две силно претоварени връзки. Оптималният маршрут в глобалния мащаб би бил да отидат на север на 01, преминавайки само през една леко претоварена връзка.



Фиг.1.13 Локално оптималното решение води до глобално не-оптимален маршрут. Пакетът се препраща от  $s = 00$  до  $d = 23$  по маршрута, подчертан в сиво. За да се избегне лекото претоварване на (01, 02), пакетът се пренасочва от 01 към 11, след което е принуден да прекоси две силно претоварени връзки

### 1.6.7 Адаптивно маршрутизиране

Адаптивният маршрутизиращ алгоритъм обикновено използва информация за състоянието на мрежата, типично дължината на опашката със заявки, това се прави с цел да се избере помежду алтернативни пътища за доставка на пакетът. Защото това маршрутизирането зависи от състоянието на мрежата, адаптивният алгоритъм на маршрутизация е тясно свързан с механизма за контрол на потока. Това се дава в контраст с детерминистичните и несъзнаващите маршрутизации, при които алгоритмът на маршрутизация и механизмите за управление на потока до голяма степен са независими и са разположени ортогонално.

### 1.25 Мъртво блокиране и живо блокиране

Мъртво блокиране възниква в мрежа за взаимно свързване, когато група агенти, обикновено пакети, не могат да постигнат напредък, защото чакат едни други да се освободят ресурси, обикновено буфери или канали. Ако последователност от чакащи агенти формира цикъл, мрежата е блокирана. Като прост пример, нека разгледаме ситуацията, показана в фиг.1.47. Връзки А и В преминават през мрежа с комутируема верига от два канала, но в момента не

могат да продължат, докато не придобият трети канал поддържана от другата връзка. Въпреки това, нито една връзка не може да освободи необходим канал от другия, докато не приключи предаването си. Връзките са блокирани и ще останат в това състояние до известна интервенция. Може да възникне застой върху различни ресурси. В този пример ресурсът е физически канал. Тя също може бъдете виртуален канал или буфер за споделен пакет.

### 1.26.1 Мъртвото блокиране

### 1.26.2 Агенти и ресурси

Агентите и ресурсите, които участват в мъртвото блокиране, се различават в зависимост от типа на управление на потока, както е показано в Таблица 1.6. За превключване на веригата, както е показано в Таблица 1.6, агентите са връзки и ресурсите са физически канали. Когато връзката е настроена, тя придобива физически канали и няма да пусне никакви от тях до завършването на връзката. Всяка връзка може неограничено да задръжте няколко канала, всички канали по пътя от източника до местоназначението.

Таблица 1.6 Агенти и ресурси който предизвикват мъртво блокиране за различни управления на потока

| Управление на потока   | Агенти | Ресурс           | Кардиналност |
|------------------------|--------|------------------|--------------|
| Превключване на вериги | Връзка | Физически медиум | Множество    |
| Пакет буфер            | Пакет  | Пакет буфер      | Единичен     |
| Флит буфер             | Пакет  | Виртуален канал  | Множество    |

### 1.26.6 Избягване на мъртвото блокиране

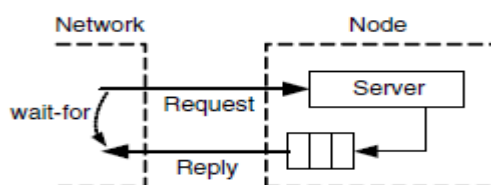
Мъртвото блокиране може да бъде избегнато чрез елиминиране на циклите в графиката за зависимостта на ресурсите. Това може да бъде постигнато чрез налагане на частична заявка на ресурсите и след това настоявайки агентът да разпределя ресурсите във възходящ ред. Затова са заблуди избягва, защото всеки цикъл трябва да съдържа поне един агент с по-висока численост ресурс, чакащ ресурс с по-нисък номер и това не е позволено от нареденото разпределение.

Докато частичен ред е достатъчен за елиминиране на циклите, и отгук и мъртвото блокиране, за простота често налагаме пълна поръчка на ресурсите номерирайки ги. Докато всички техники за избягване на мъртво блокиране използват някаква форма на подреждане на ресурсите, те се различават по начина, по който ограниченията, наложени от това нареждане на ресурсите, засягащи маршрутизацията. При някои подходи ресурсите могат да бъдат разпределени без ограничение по маршрута. При други подходи броят на

необходимите ресурси се намалява за сметка на забрана на някои маршрути, които иначе биха нарушили ресурса заявки. [13, 16]

### 1.26.5 Мъртво блокиране на високо ниво (протокол)

Мъртво блокиране може да бъде причинено от зависимости извън мрежата. Например, разгледайте случая, показан на фигура 1.52. Най - горният мрежов канал чака сървър за премахване на заявен пакет от мрежата. Сървърът от своя страна е ограничен с буфериране и по този начин не може да приеме пакета за заявка, докато подолен канал не приеме пакетът и отговори на изискването от изходния буфер на сървъра пакета да бъде приет. Всъщност горният канал чака на долния канал поради външния север. Този край на графиката за изчакване е това което се дължи не на самата мрежа, а на сървъра. Мъртвото блокиране причинено от изчакващи цикли които включват такива външни краища, често се нарича мъртво блокиране на високо ниво или мъртво блокиране на ниво протокол.

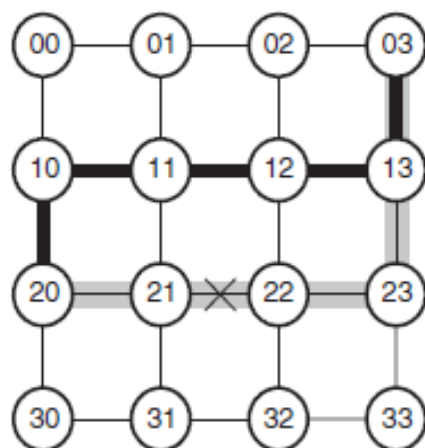


Фиг.1.52 - Имплицитна зависимост на ресурси в система за заявка-отговор

### 1.29 Живо блокиране

За разлика от мъртвото блокиране, пакетите, които са живи, продължават да се движат в мрежата, но никога не достигат до местоназначението си. Това е преди всичко грижа за не-минималния маршрут алгоритми, които могат да препратят погрешно пакетите. Ако няма максимална гаранция от време, пакетът може да бъде неправилно зададен и може да остане в мрежата за неопределено време. Техниките за управление на потока също може да доведат до живо блокиране. Ако има пакет, който се отхвърля всеки път, когато отново навлезе в мрежата, може никога да не достигне целта си.

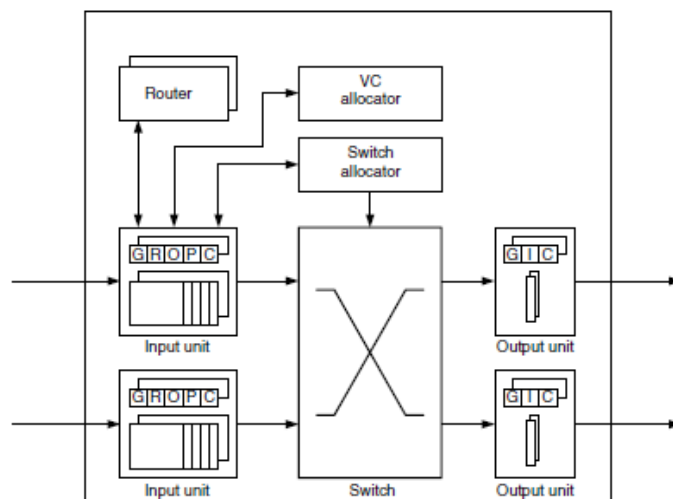
Има две основни техники за избягване на живото блокиране: детерминистично и вероятно избягване [15].



Фиг.1.64 - Отношението на допускане за повреда при разрешаване на първоначален скок преди маршрутизиране по посока на посоката започва (само за простота се използват две измерения)

### 1.30.1 Основна архитектура на маршрутизатора

### 1.30.2 Блокова диаграма



Фиг.1.65 Блокова диаграма на типичен маршрутизатор с виртуален канал [10, 62]

Тъй като в тази глава разгледахме необходимите алгоритми за маршрутизация от трите най-големи групи и това:

- Детерминиран алгоритъм на маршрутизация;
- Несъзнателни алгоритми на маршрутизация;
- Адаптивни алгоритми на маршрутизация.

След това следва анализ на мъртво и живо блокиране на мрежата при съответна топология.

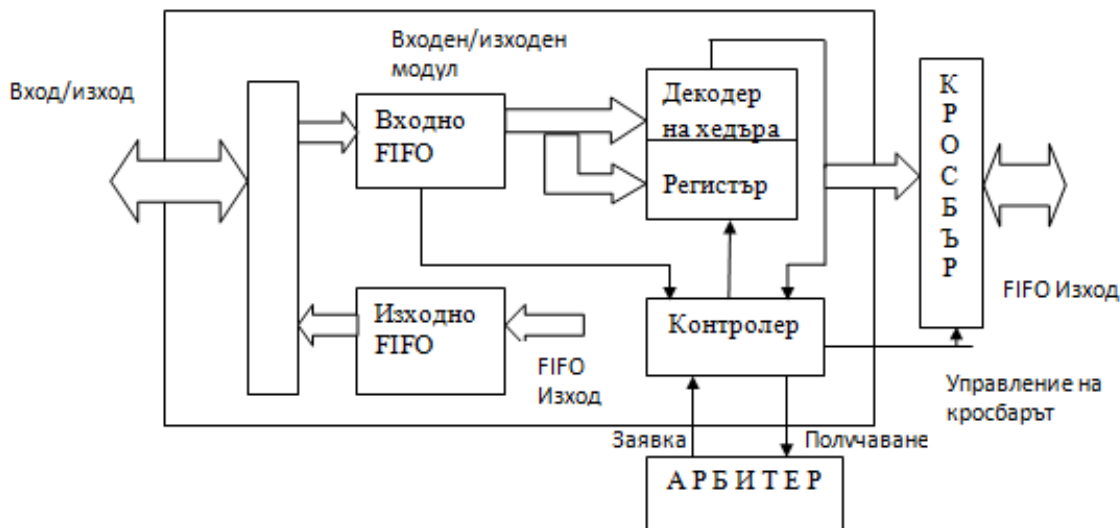
Разглеждаме структура на мрежов маршрутизатор дадена на фиг. 1.65 с редуцирана латентност.

## Изводи от Глава 1

Следва да се използват и трите най-големи групи на маршрутиращи алгоритми със съответните подгрупи при оценката на мрежовите топологии на решетка и торус при различни модели на трафик, дадени в Таблица 1.1. От тук се получава оценка за поведението на мрежата при различни модели на трафик и различни маршрутиращи алгоритми, които след това ще ги приложим при мрежите на чип (МнЧ). Даваме начална точка за блокова схема на маршрутизатор, която следва да се използва при мрежите на чип (МнЧ)

## ГЛАВА 2. Избор на техническо решение за смарт суич. Акценти. Блокова Схема. Интерфейси

Върху разгледаната в глава 1 проблематика следва да се предложи техническо решение и блокова схема на маршрутизатор. Върху направените изводи от глава 1 се предлага блокова схема на маршрутизатор, която следва да се приложи, като обобщената схема е дадена на фиг. 2.1.



Фиг.2.1 - Блокова схема на суича

Главните акценти върху тази предложена структура са дадени върху две части и това:

- Латентност
- Производителност

Тъй като вече разгледахме в глава 1 това, което следва да се приложи тук в глава 2, при проектирането на блоковата схема е необходимо да се увеличи производителността при следните условия:

- Избиране на високоскоростни интерфейси за свързване с увеличена разрядност на шината за предаване на данните, например XAUI (10 Gigabit Attachment Unit );

- Избиране на високоскоростен процесор, изпълняващ ролята на маршрутизиращ суич;
- Добавяне на високоскоростни памети за FIFO (входяща и изходяща памет), най-често статична високоскоростна SRAM памет с увеличен брой на цикли четене / запис.

Намаляване на латентността при следните условия

- Използване на по-малка опашка за заявки, т.е. по малки FIFO за заявки;
- Използване на по-малки буфери на вход и изход за буфериране на пакетите, т.е. по малка дълбочина на FIFO за данни;
- Използване на високоскоростен процесор от модерна генерация UltraScale+ (L=16nm) технология на Xilinx или подобен с малко собствено закъснение;
- Използване на по-малко закъснение при предаване и приемане както протоколите, така и касаещи подобни приемно-предавателни модули, например SFP+ за достигане скорости до 10Gb/s.

Ако самото приложение изисква увеличена производителност или пък увеличена пропускателна способност можем да използваме съответно и високоскоростни интерфейси и интерфейси с висока пропускателна способност. За нашата цел на проектиране на суича, сме подбрали високоскоростни и с малка латентност интерфейси, това което използваме е интерфейс 10Gbps, свързан към съществуващ XAUI протокол.

За да се осъществи висока скорост на предаване на данните, този протокол ще ни даде максимално 10 Gb/s, т.е. пълната пропускателна способност няма да надхвърля 10 Gb/s. Включително към XAUI интерфейса са включени 4 x 3.125 Gb/s, но самата скорост на предаване от теоретична в нашия практически случай не може да е по-голяма. Това е тествано с модул, съществуващ чрез използване на протокола резултата излиза че до към 10 Gb/s ние успяваме да постигнем максимално. Това, което е необходимо за да се получи пропускателна способност от порядъка на двойно повече предаване е чрез използване на вътрешна шина с разреденост от 64-бита на FPGA, като на него да се реализира XAUI ядрото, което се свързва към медийния слой, или както е известен още като машинен MAC слой. Използването на вътрешната скоростна линия за предаване на данни е идеално решение, но не всеки път. При необходимата гъвкавост за предлагане на не вече заложен, а друг маршрутиращ алгоритъм, ние губим ценни ресурси в дизайна. Значи, ако искаме да получим и голяма пропускателна способност, ние трябва да имаме или високоскоростна шина с по-малка разреденост - примерно 32 бита или пък ако скоростта се намали,

тогава трябва да търсим шина която е с по-голяма широчина, или 64 битова шина.

Друго, което се взима в предвид тук е самото използване на колкото е възможно по-малко разстояние на свързване между източника и самия интерфейс. В случая ние подбираме XAUI – външен интерфейс със съвместим за него приемащо-предаващ модул.

В нашия случай ние имаме кабел който отговаря на изискванията за близка комуникация от 1м и може да е използва за обмен на данни свързан към XAUI интерфейса който да е външен за FPGA-то или пък възможно е всичко това да върви в едно, или както е казано XENPAC модул. Значи ако искаме да осъществим висока скорост добре би било да подберем по едни модул свързан направо към отделно FPGA и възможно осъществяване на връзка по високоскоростна шина с друго FPGA. Това изисква използване на останалите x4 Gen 2 PCI-e или шина която е доста известна напоследък в нашия случай скоростта постигната чрез този протокол дава закъснение от към 5ns и пропускателна способност от почти 20Gb/s. Това, което е налично при нас е използване на приемно-предавателни модули, но със скорост от не повече от 10Gb/s или ако имаме два модула съответно тогава не може пропускателната способност да е повече от 20Gb/s. Добре е останалите приемно-предавателни модули вместо по шина PCI-e да се свържат към друг интерфейс на друго FPGA. Това, което е известно е че по-новите поколения FPGA съдържат и по-високоскоростни интерфейси за комуникация или GTN, GTY и GTM, докато старите интерфейси GTX вече обяснихме че не може да са повече от 3.125Gb/s. Докато тези нови интерфейси достигат почти до 50 Gb/s GTN, GTY и GTM.

По отношение на FIFO паметта ние предлагаме използване на високочестотна памет с подобно използване на широка пропускателна способност. Това, което трябва да се използва за да няма задръстване още в самото начало при пристигане на пакета или съответно хедъра е използване на QDR (Quad Data Rate) или по-точно памет с висока скорост и широка пропускателна способност. Точно е че тази памет се използва само при специализирани уреди, но също така следва да се каже че използването на DDR (Double Data Rate) ще доведе до значително закъснение при поява на пакет за обработка или пък флит. Получаването на скоростта също така трябва да бъде отразено и от самата логика, сега ако вземем предвид че използване на специализиран чип (ASIC) ще е по-добре, но дял от настройките ще липсват, или пък е необходимо да се вградят в него всички маршрутиращи алгоритми за да можем да постигнем изискваната цел. Това което се направи е подбор на FPGA чип и към това можем да вземем в предвид всички нови поколения на Ultrascale+ на Xilinx или пък от някой друг производител с подобни характеристики. Когато вече сме решили проблема с FPGA чипа, тогава следващата стъпка е проектиране на заложената функция маршрутиращ алгоритъм и другите модули показани на фиг.2.1 Това, което досега бе проектирано по самия маршрутизатор е част от блоковата схема. Ускорението на системата е също така добре дошло, но въпреки това ние предлагаме вместо

ускорение да се използва синхронизация между FIFO паметта и контролера за обработка на пакетите и флитове, съответно.

## **Изводи от Глава 2**

Тъй като вече е предложена съответно блоковата схема, дадена на фиг.2.1, следва да се подбере платформа, която да отговаря на по-горе дадените изисквания:

- UltraScale+ FPGA чип на 16nm технология;
- FIFO, съставено от QDR (Quad Data Rate) памет с удвоена два пъти скорост от DDR (Double Data Rate) памет;
- Високоскоростни и с малка латентност протоколи за предаване и приемане на данни XAUI;
- Интерфейси за външна комуникация - най-добре би било медийния слой да е извън FPGA чипа, т.е. отделна специализирана разработка на друг чип, и да използва SFP+ (Small Fiber Pluggable) оптични свързващи интерфейси със скорост до 10Gb/s.

## **ГЛАВА 3. Получени резултати от симулации и сравнение на резултатите от използваните топологии**

### **3.1 Получени резултати от симулации при торус топология**

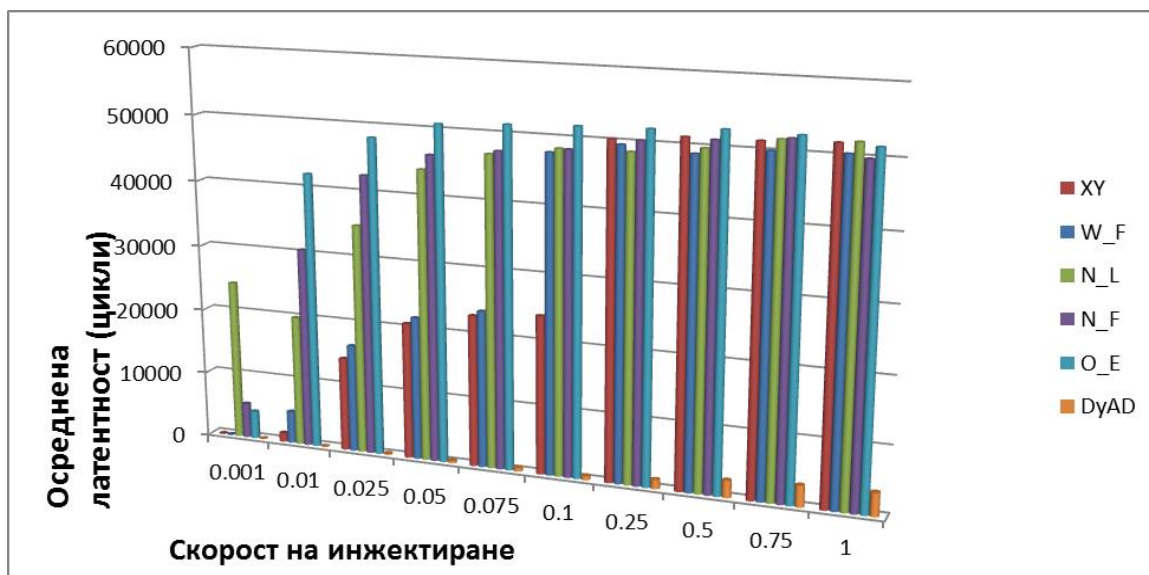
В досегашното изследване се дадоха съответно предложение и се изследваха решения за суичове, реализирани на МнЧ. Тези изследвания ни дават представа за самите характеристики на суичове, реализирани върху МнЧ и най-вече производителността и закъснението.

Резултатите от първото изследване са дадени в таблица 3.1. В самата таблица са представени всички параметри, които предоставя симулатора Noxim [33] и е необходимо да се дадат сравнителни резултати и да извлекат съответните изводи.

Това, което използваме тук като входни данни на симулатора са маршрутиращите алгоритми и тук само ще ги споменем, повече детайли по тях са разгледани в глава 1.

- XY – първо по X след това Y маршрутизиращ алгоритъм;
- W\_F - първо обръщане към запад маршрутизиращ алгоритъм;
- N\_L – първо обръщане към север маршрутизиращ алгоритъм;
- N\_F – негативно обръщане първо маршрутизиращ алгоритъм;
- O\_E – нечетно, четно обръщане маршрутизиращ алгоритъм;
- DyAD – динамично адаптивен детерминиран алгоритъм.

Получените резултати са дадени съответно на графиката на фигура 3.1, която изследва зависимостта на усреднената стойност от скоростта на инжектиране на пакетите.



Фиг.3.1 Зависимост на усреднената латентност изразена в цикли от скорост на инжекция на пакетите

От фигурата 3.1 се вижда, че при малка инжекционна скорост в мрежата на МнЧ се получава малка латентност за почти всички алгоритми на маршрутиране. Но почти всички следват едно и също закъснение. Това, което може да се каже тук е, че латентността може да се увеличи само при скорост на инжектиране, която е малка и всички следят, но въпреки това става ясно че латентността се следи от всички алгоритми, но заключаваме че както беше обяснено в глава 2 само XY алгоритъма е с най-малка латентност което се продиктува от неговите възможни стъпки. След 0.1 инжекционна скорост виждаме че рязкото нарастване на латентността става видимо от графиката на фиг.3.1.

Поради последователността на стъпките, видимо е че имаме също така и малка стойност на латентността за W\_F алгоритмът на маршрутизация, но това е така до стойност 0.075, значи следователно имаме рязко нарастване на латентността.

При N\_L можем да се види, че поради последователността на стъпките при маршрутизация заключаваме че латентността е малка, но само при стойност от 0.01, което показва че скорост е малка.

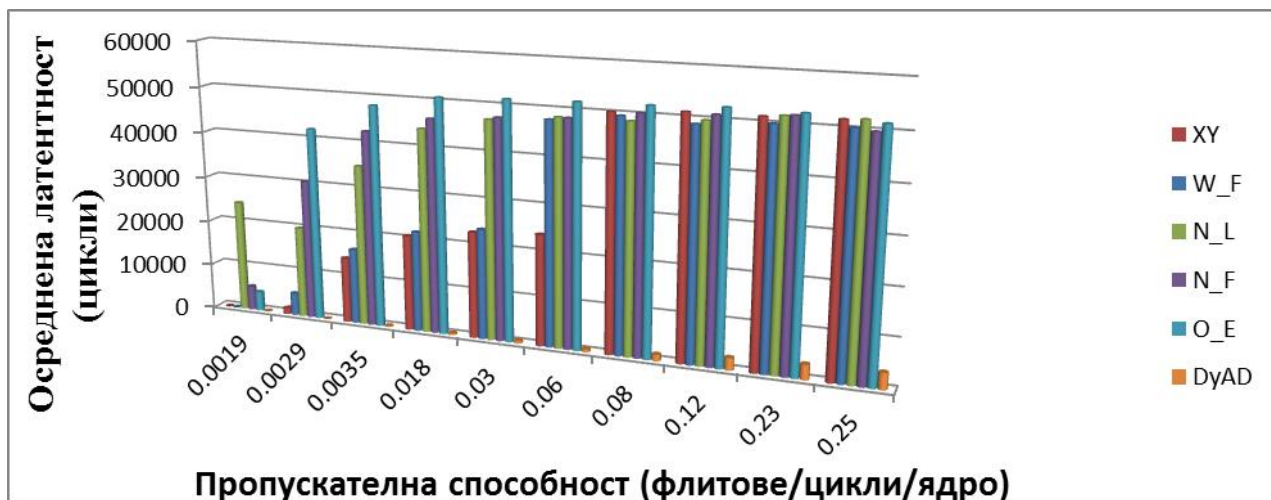
При N\_F при това положение се получава малка латентност, но при скорост на инжектиране 0.001, което не е добре ако искаме да използваме този маршрутизиращ алгоритъм.

При O\_E важно е да се каже че стойността на латентността се потвърждава че е най-висока спрямо другите маршрутизиращи алгоритми.

DyAD алгоритмът е най-добрата опция в случая, при който получаваме най-ниска стойност на латентността. Това се дължи на самия факт че алгоритмът използва динамично адаптивно и детерминистично свойство. Така че до определено ниво алгоритмът използва свойството на случайно маршрутиране чрез намиране на най-малкия път, след това може да стане гъвкав и може да стане адаптивен и да използва информация за състоянието на мрежата чрез

използване монитори в мрежата за адаптивно маршрутиране, както е показано на фиг.3.2.

Докато на фиг. 3.3 можем да видим зависимостта на средната латентност от пропускателната способност.

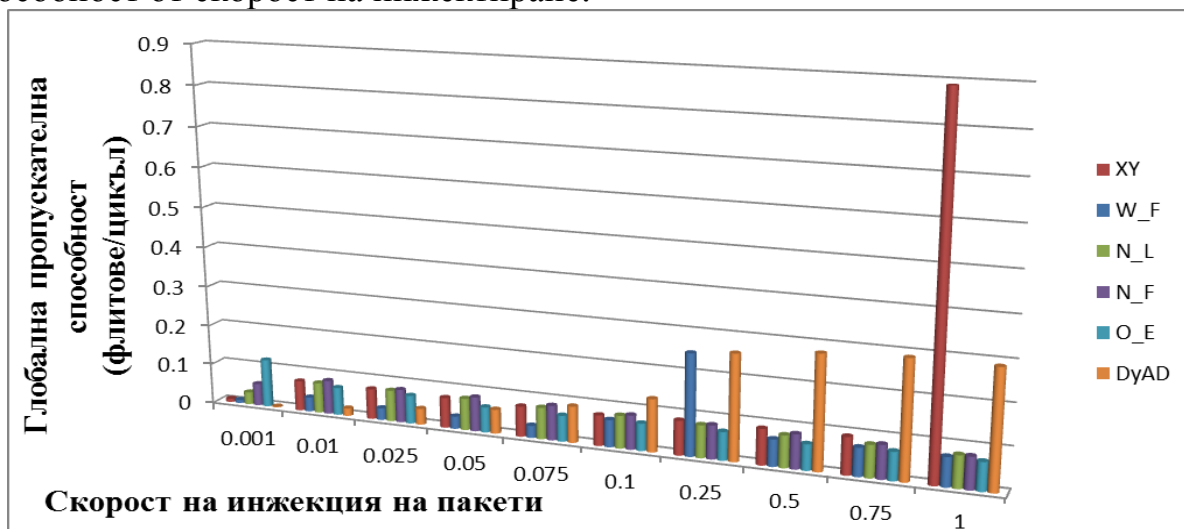


Фиг.3.3 Зависимост на средната латентност в зависимост от пропускателната способност

От фигура 3.3 е видимо, че получаваме добра пропускателна способност за всички маршрутиращи алгоритми, но можем да кажем че все пак най-добре ще е използването на XY – алгоритъм, тъй като осигурява сравнително високо ниво на пропускателна способност дори и при висока стойност на закъснението.

Добре е това, че при DyAD алгоритъма получаваме и най-ниската стойност на закъснение при най-висока стойност на пропускателната способност.

На фигура 3.4 можем да видим зависимостта на Глобалната пропускателна способност от скорост на инжектиране.



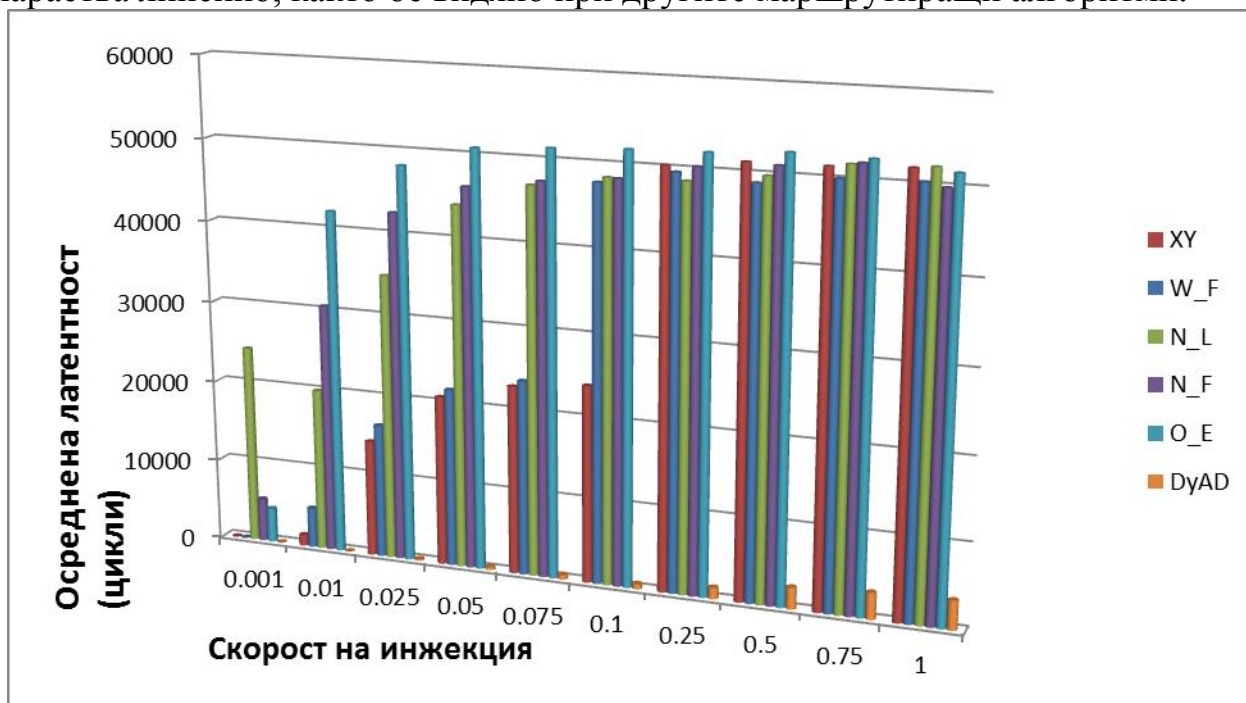
Фиг.3.4 Зависимост на глобалната латентност от скоростта на инжекция

При анализа на фигура 3.4 може да се заключи, че получаването на най-висока пропускателна способност при най-висока скорост на инжектиране на пакетите може да получи само с XY алгоритма. Това показва, че XY алгоритмът може да се използва в МнЧ суичове.

### 3.2 Получени резултати при топология решетка

На този етап ще оценим топологията решетка и ще получим данни за получаване на изводите за важните параметри. В досегашното изследване съответно дадохме предложения и изследвахме решения за суичове, реализирани на МнЧ. Тези изследвания ни дават представа за самите характеристики на суичове, реализирани върху МнЧ комутатор и най-вече по отношение на производителността и закъснението.

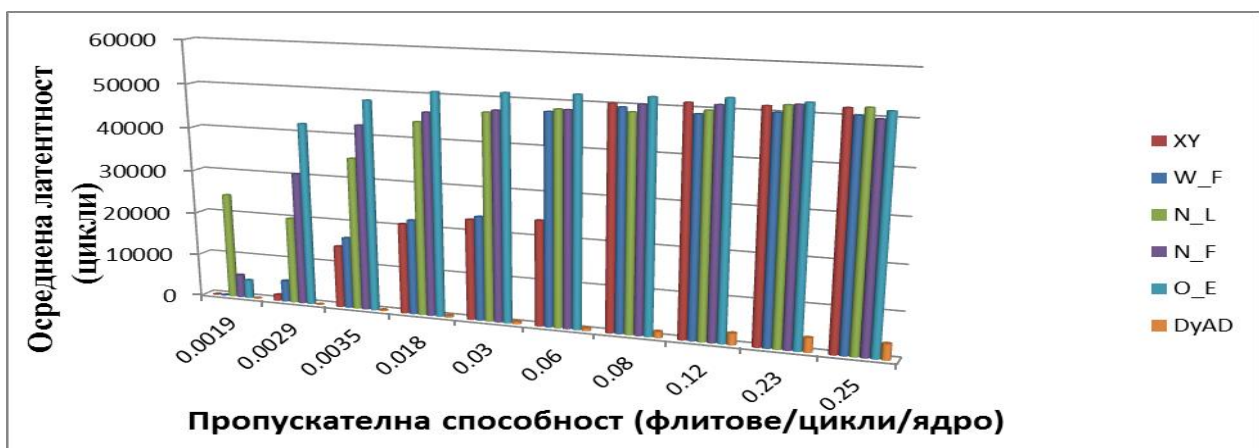
Това, което е заключение от таблица 3.2, както и от фигура 3.5 е ясно че използването на алгоритъма XY все още дава най-добра латентност, въпреки че издържа и най-голяма инжекционна скорост. Малко с по-лоши качества е W\_F алгоритмът, но той след стойност на инжектиране от 0.05 рязко вдига закъснението почти двойно, все пак тази характеристика се дължи на използваните стъпки които алгоритъма осъществява при маршрутирането. Най-добра характеристика при използване на динамично адаптивен детерминистичен алгоритъм (DyAD) е все пак с най-добра крива на натоварване, въпреки това можем да видим че нарастването не става рязко а нараства линейно, както бе видяно при другите маршрутиращи алгоритми.



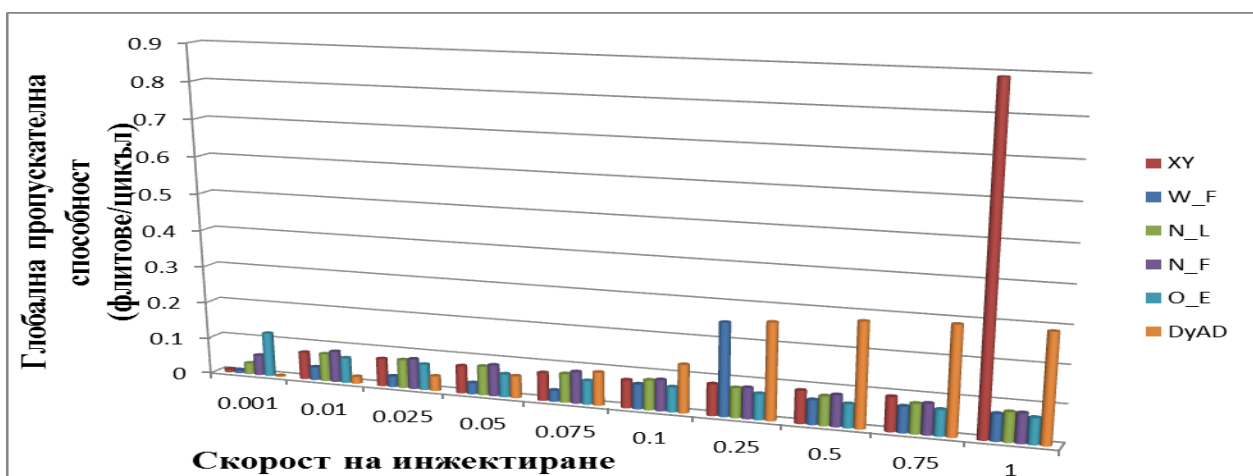
Фиг.3.5 Показана зависимост на усреднената латентност от скоростта на инжекция

При фигура 3.6 пък може да се забележи характеристиката на латентността в зависимостта от пропускателната способност. Тук сме заключили, че зависимостта само при XY алгоритъма на маршрутиране става много добра, докато при алгоритмът динамично адаптивен детерминистичен (DyAD) е с най-малка стойност на латентността и това става така че използваме монитори за състоянието в мрежата. На фигура 3.7 е начертана характеристиката на глобалната пропускателна способност от скоростта на инжекция - от тук можем да видим как нарастването на глобалната пропускателна способност от инжекцията на пакетите, и това по следната последователност.

За XY – маршрутиращия алгоритъм нараства но все пак показва качества до относително висока глобална пропускателна способност от 0.06, след това следва W\_F алгоритмът с глобална пропускателна способност от 0.06, и накрая имаме глобална пропускателна способност от 0.25 на при (DyAD). От тук можем да кажем че (DyAD) алгоритмът се държи добре при нарастване на скоростта на инжекция на пакетите в мрежата, така и на увеличена пропускателна способност.



Фиг.3.6 Показана зависимост на усреднената латентност от пропускателна способност



Фиг.3.7 Показана зависимост на глобална латентност от скорост на инжекция

Това, което можем да заключим от фигура 3.5, фигура 3.6 и фигура 3.7 е че при топологията решетка, която е по-проста за осъществяване че нямаме връзки от крайните възли към крайните възли на маршруторите има само един проблем а това не е недостатъчната пропускателна способност спрямо торус топологията.

### **3.3 Изследване на маршрутиращите алгоритми по отношение на енергийната консумация**

Тук са представени данни от изследванията които са осъществени за да направим оценка на изразходваната енергия при двата маршрутиращи алгоритъма, които са с най-качествени показатели и това са малка латентност и голяма пропускателна способност, т.е XY и DyAD алгоритми. За които ще направим допълнителни изводи, дадени в глава 4, тъй като те ще са използват в работата със схемата.

#### **3.3.1 Симулационните сценарии което ще го използваме тук е базирано върху изследване на торус 7 X 7 топология и това за трафик с изместващ се бит за предложените маршрутиращи алгоритми.**

1. Използване на торус топология с 7 X 7 възли за получаване на резултати от усреднената пропускателна способност и средната латентност при изместващ се бит трафик модел, при пакет с фиксирана дължина и променлива скорост на инжектиране на пакетите. Използване динамично адаптивен детерминистичен алгоритъм на маршрутиране (DyAD) дадено на фигура 3.10.
2. Използване на торус топология с 7 X 7 възли за получаване на резултати от глобалната средната пропускателна способност и усреднената латентност, при изместващ се бит трафик модел, при пакет с фиксирана дължина и променлива скорост на инжектиране на пакетите. Използване на алгоритъм XY, дадено на фигура 3.11.
3. Използване на торус топология с 7 X 7 възли за получаване на резултати за усреднената пропускателна способност и изразходвана енергия при изместващ се бит трафик модел, при пакет с фиксирана дължина и променлива скорост на инжектиране на пакетите. Използване динамично адаптивен детерминистичен алгоритъм на маршрутиране (DyAD) и XY дадено на фигура 3.12.

На фигура 3.8 и фигура 3.9 съответно са дадени снимки на получените резултати от самия симулатор.

```

tokic@ubuntu: ~/noxim/bin
File Edit View Terminal Help
amples/other/config16x16.yaml

SystemC 2.3.1-Accellera --- Oct 27 2015 09:35:25
Copyright (c) 1996-2014 by all Contributors,
ALL RIGHTS RESERVED

Noxim - the NoC Simulator
(C) University of Catania

Loading configuration from file /home/tokic/noxim/config_examples/other/co
nfig16x16.yaml
Reset...
done! Now running for 10000 cycles...
Noxim simulation completed.
( 11000 cycles executed)
% Total received packets: 2329
% Total received flits: 4658
% Global average delay (cycles): 23.3963
% Global average throughput (flits/cycle): 0.00203952
% Throughput (flits/cycle/IP): 0.0020217
% Max delay (cycles): 66
% Total energy (J): 6.66083e-05
%   Dynamic energy (J): 3.74341e-07
%   Static energy (J): 6.6234e-05
tokic@ubuntu:~/noxim/bin$

```

```

tokic@ubuntu: ~/noxim/bin
File Edit View Terminal Help
amples/other/config16x16.yaml

SystemC 2.3.1-Accellera --- Oct 27 2015 09:35:25
Copyright (c) 1996-2014 by all Contributors,
ALL RIGHTS RESERVED

Noxim - the NoC Simulator
(C) University of Catania

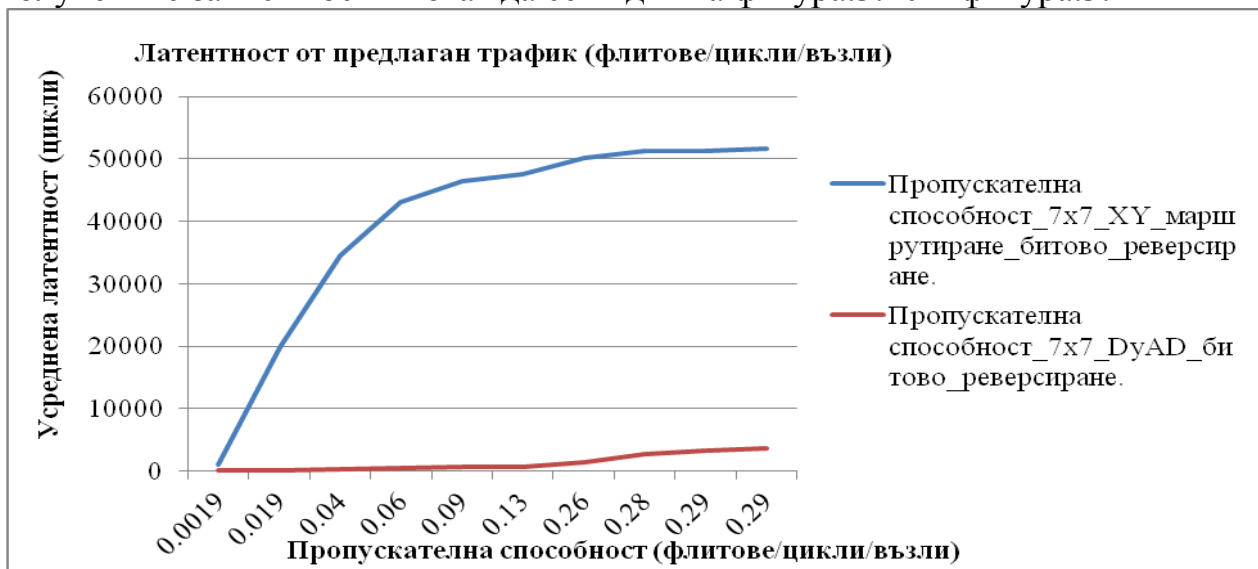
Loading configuration from file /home/tokic/noxim/config_examples/other/co
nfig16x16.yaml
Reset...
done! Now running for 10000 cycles...
Noxim simulation completed.
( 11000 cycles executed)
% Total received packets: 2302
% Total received flits: 4607
% Global average delay (cycles): 23.2042
% Global average throughput (flits/cycle): 0.00201024
% Throughput (flits/cycle/IP): 0.00199957
% Max delay (cycles): 68
% Total energy (J): 6.66043e-05
%   Dynamic energy (J): 3.70367e-07
%   Static energy (J): 6.6234e-05
tokic@ubuntu:~/noxim/bin$

```

Фиг.3.8 - При използване на XY алгоритъм      Фиг.3.9 - При използване на DyAD алгоритъм

### 3.2 Оценка на данните от осъществените резултати

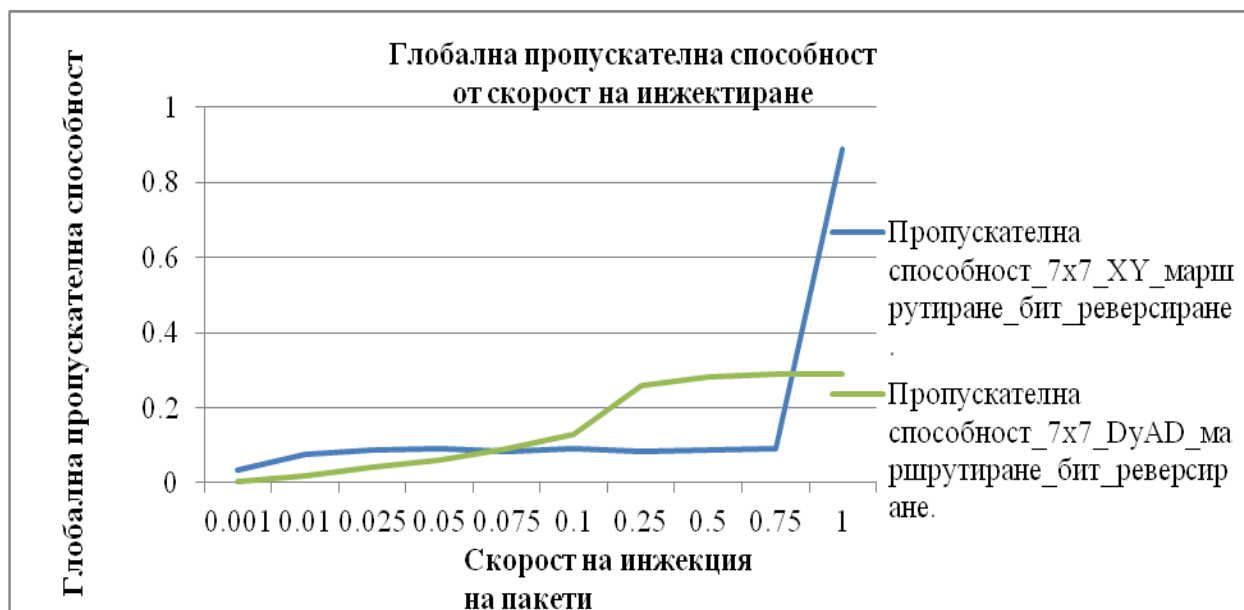
Получените зависимости могат да се видят на фигура.3.10 и фигура.3.11



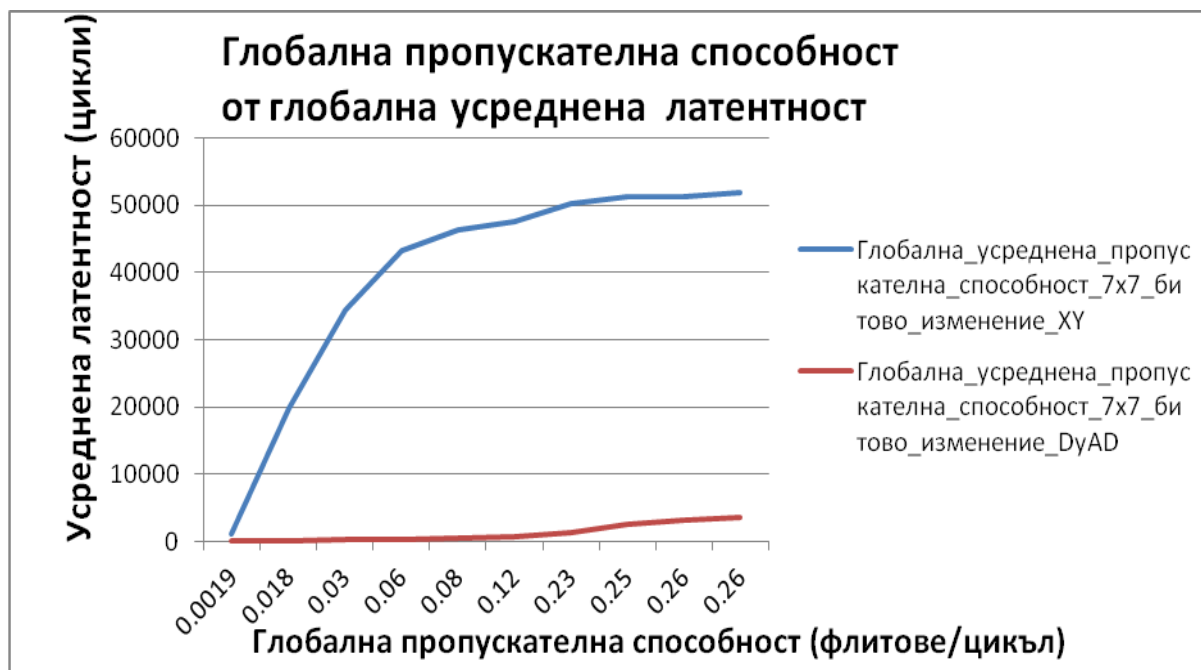
Фиг.3.10 Зависимост на латентността от предлаган трафик за мрежа 16 x 16 XY маршрутиране XY и за DyAD

От дадената зависимост можем да видим, че при динамично адаптивния алгоритъм (DyAD) средната латентност е доста добра при висока скорост при предлаган трафик изменящ се бит.

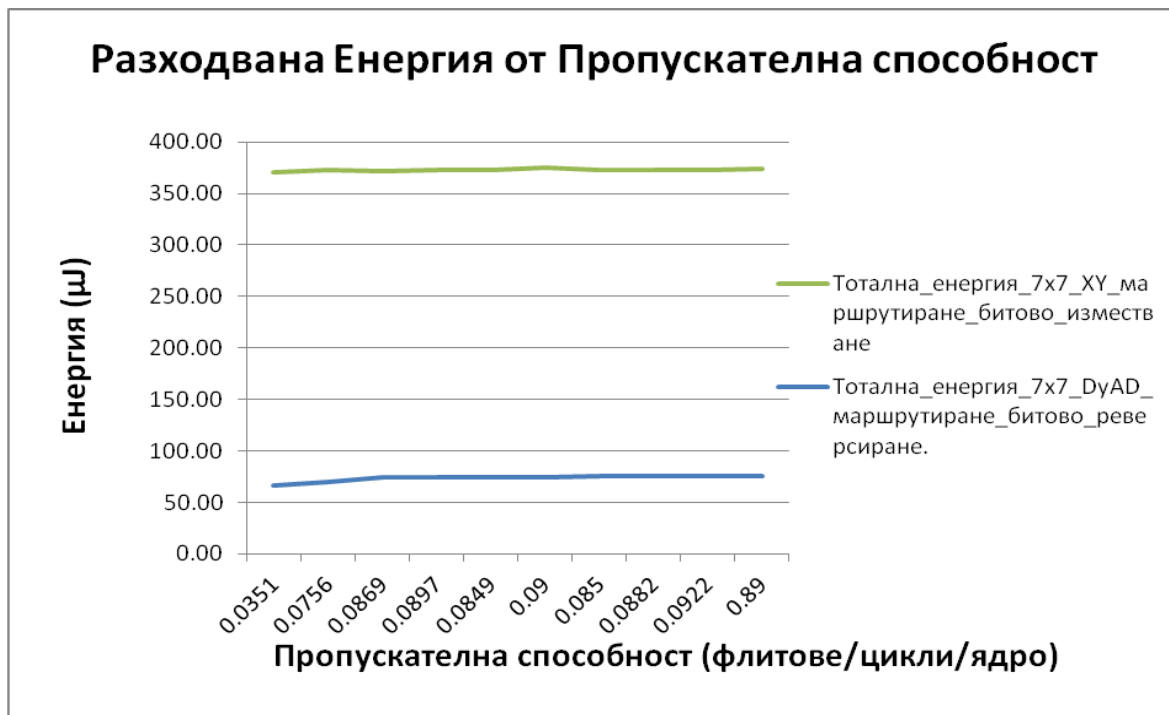
За зависимост на алгоритъма можем да кажем че това, което е дадено тук, отговаря на изискванията за XY но въпреки това има висока стойност на латентността при натоварване на мрежата, което е от порядъка на 50-60 хиляди цикли.



Фиг.3.11 - Зависимост глобална пропускателна способност от скорост на инжекция за мрежа 7 x 7 XY маршрутиране XY и за DyAD



Фиг.3.12 - Зависимост на латентността от глобалната пропускателната способност за мрежа 7 x 7 XY маршрутиране XY и за DyAD



Фиг.3.13 - Зависимост на разходваната енергия от глобалната пропускателната способност за мрежа 7 x 7 XY маршрутиране XY и за DyAD

Това, което е видимо от фигура 3.11 и е добре да се забележи е, че пропускателната способност на XY е доста по-висока от тази на DyAD, обаче проблема е там че имаме и доста висока латентност за сметка на това. Зависимостта на глобалната средна пропускателна способност от броя цикли може да се види от фигура 3.12, при което също така е очевидно че получаваме малка стойност на закъснение от пропускателната способност за DyAD спрямо XY алгоритъма.

На фигура 3.13 получаваме една от много значимите характеристики и това е стойността на изразходваната енергия от пропускателната способност. От тук е видимо, че стойността на изразходвана енергия е поне с приблизително 10 пъти по-малка за DyAD спрямо тази при XY.

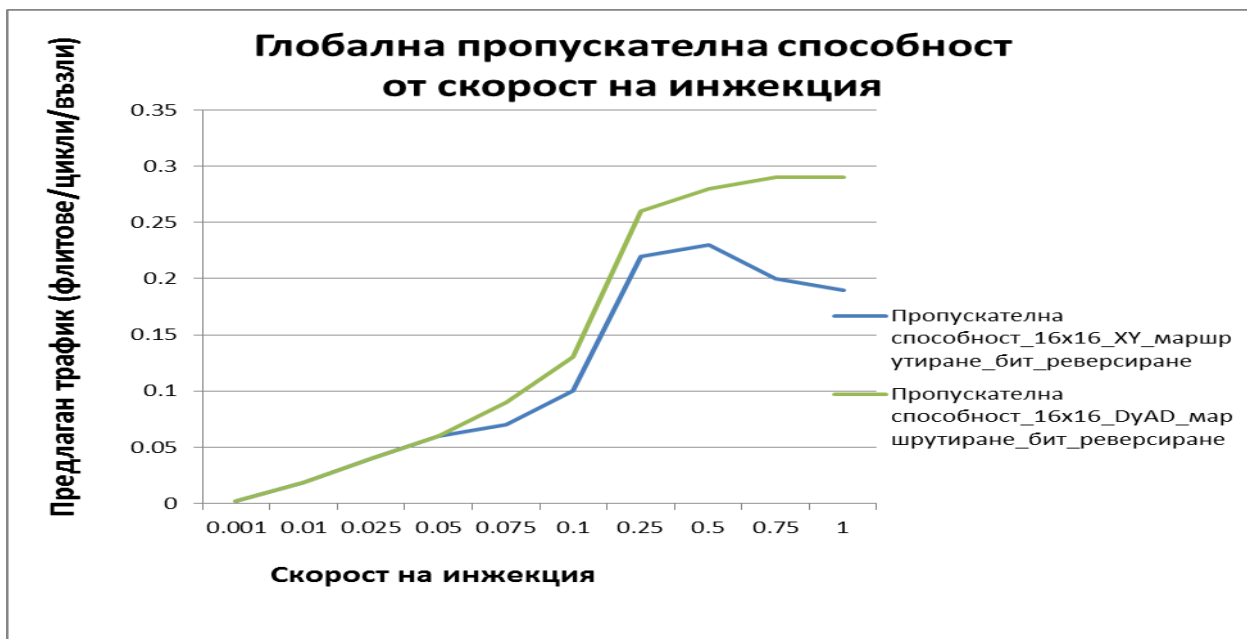
### 3.4 Изследване на торус топология с 16 x 16 възли

Изследване на торус топология с 16 x 16 възли за получаване на качествени параметри и оценяване на същата.

Даваме съответно отбелязване на резултатите от получени от настройката на симулатора Noxim за мрежата чрез използване на XY маршрутиращ алгоритъм, така и за DyAD маршрутиращ алгоритъм.

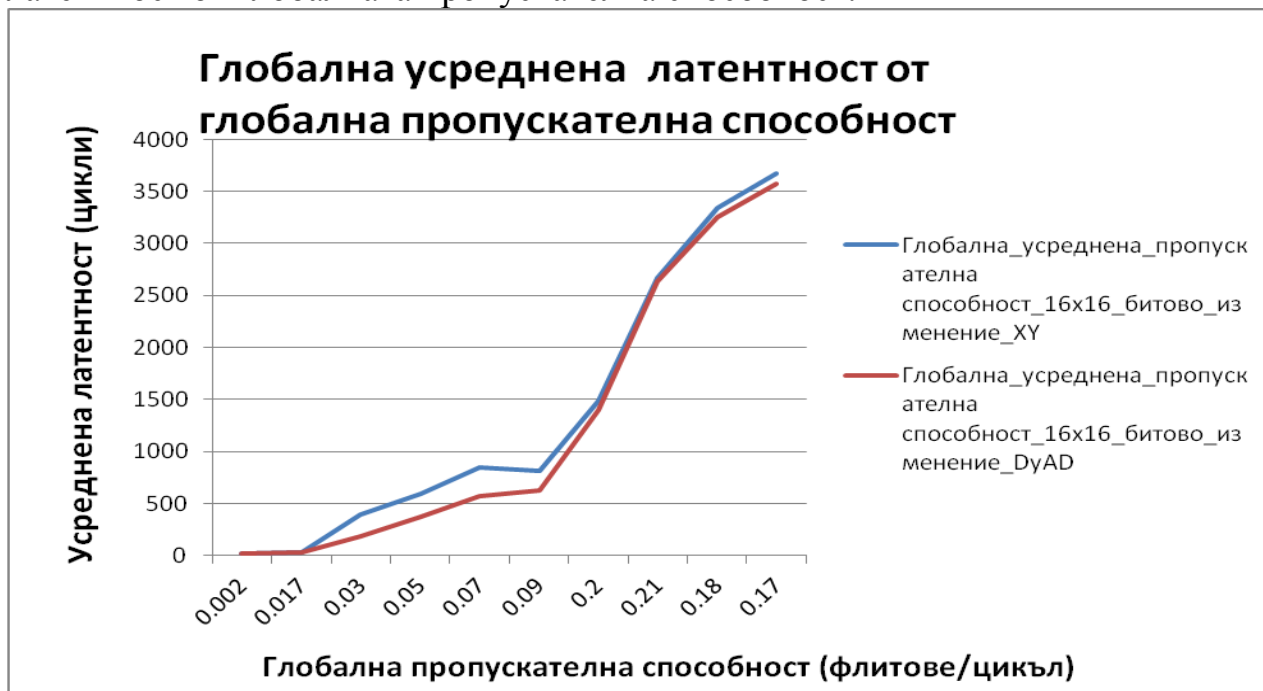
На фигура 3.14 е показана зависимостта на латентността от пропускателната способност за 16 x 16 торус. Това кое е видимо от тази фигура е че латентността е малка но също веремено и достатъчно помалка при използване на DyAD алгоритъмът. Порядъка на закъснението е до приблизително към 4 хиляди цикъла и за двата алгоритъма. Това което получаваме е че можещи да

закъснението е също но сега нека да видим и зависимостта на пропускателната способност от инжектирания трафик представено на фигура 3.14. Можем да забележим, че скоростта на инжектирания пакет и пропускателната способност е доста по-висока за DyAD алгоритмът.



Фиг.3.14 - Зависимост на разходвана енергия от глобалната пропускателната способност за мрежа 16 x 16 за маршрутиране XY и за DyAD

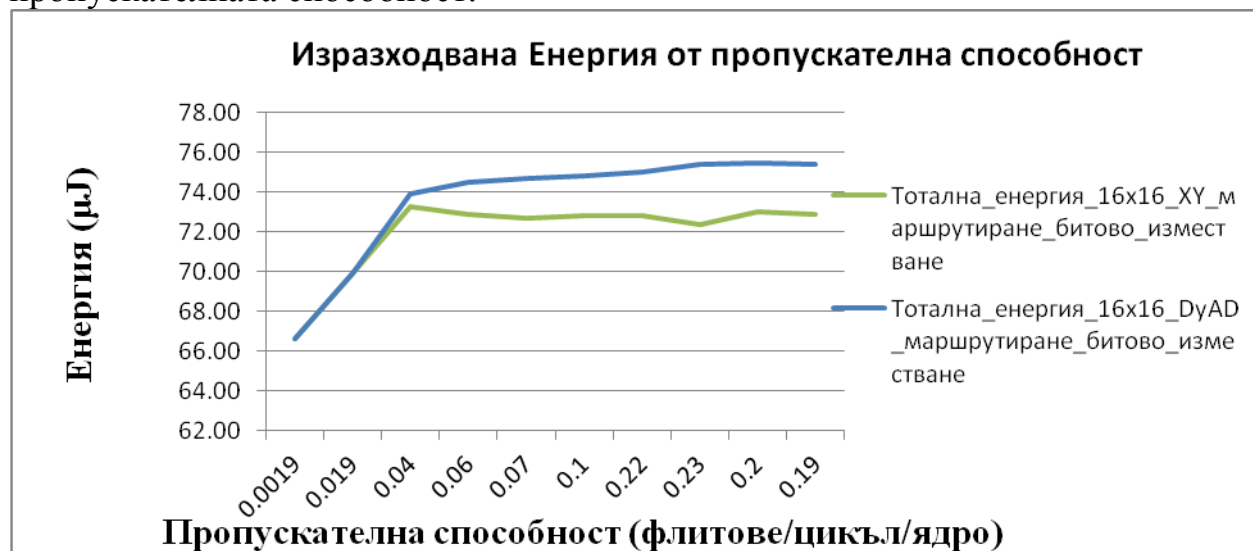
Докато пак от фигура 3.15 можем да видим зависимостта на глобалната латентност от глобалната пропускателна способност.



Фиг.3.15 - Зависимост на средна латентност (цикли) от глобалната пропускателната способност за мрежа 16 x 16 за маршрутиране XY и за DyAD.

Разсъждайки по фигура 3.15 можем да се заключи, че до определено ниво DyAD има по-малка почти с половина от усреднена латентност по-ниска при една и съща пропускателна способност.

На фигура 3.16 имаме зависимостта на изразходваната енергия от пропускателната способност.



Фиг.3.16 – Зависимост на изразходваната енергията от глобалната пропускателната способност за мрежа 16 x 16 за маршрутиране XY и за DyAD.

От фигура 3.16 пак е видимо че имаме около 0.20 пъти по-малко изразходвана енергия при еднаква пропускателна способност при DyAD спрямо XY алгоритмът.

### Изводи от глава 3

Изследвани бяха топологии тип решетка и торус по отношение на:

- При два типа на трафик и това бит изменящ, и случаен;
- Получена е качествена оценка за латентността на всяка топология по отделно;
- Получена е качествена оценка за пропускателната способност и скоростта на инжекция за всяка структура;
- Получена е качествена оценка за маршрутизиращите алгоритми по отношение на енергийна ефективност, при взимане различна стойност на пропускателната способност

## ГЛАВА 4. Изводи при избирането на подходящ маршрутиращ алгоритъм и топология за използване при МнЧ суич проектиране. Перспективи. Приложения

Тъй като вече в Глава 3 сме получили качествени резултати от изследването на топологията, както и на маршрутиращ алгоритъм подходящ за тази топология, в тази глава ще се върнем на тези резултати и ще ги използваме за да получим най-важните изводи в тази дисертация.

Тъй като много важен показател е топологията, най-важно е да започнем с обсъждане на нея. В зависимост от достъпните ресурси за прототипизиране и крайната цел на използване на този тип маршрутизатор предлагам следното ако платформата е почти ограничена с ресурси тогава използваме архитектура решетка (Mesh) това е така че при нея са изоставени крайните свързващи линии на крайните маршрутизатори. Това е предимство но можем да кажем че при подбиране на тази топология губим ценна пропускателна способност. Също така можем да заключим, че при използването на топология решетка 7x7 можем и да получим висока усреднена латентност, което е показано в таблица 3.1, както и на фигура 3.1 в зависимост от скоростта на инжекция, или на фигура 3.3 в зависимост от пропускателната способност.

Използване на топология торус е при наличие на необходимо достатъчно ресурси при прототипизирането - в този случай линиите които свързват от край до край на маршрутизаторът довежда до получаване на ефекта на получаване на усреднен път, тъй като пакета не мора да минава през вътрешния канала ако е зает ами ще се пренасочи през външния и тогава за него този път ще е на половина от целия път подробно обяснено в Глава 1. Това е много добре при тази топология, също така разсъждайки за латентността може да се види от резултатите получени в таблица 3.1 както и при сравнение на резултатите при двата маршрутиращи протоколи че латентността е малка. Това се дължи предимно на свойството че този алгоритъм управлява чрез потока с доста особен приспособяващ адаптивен алгоритъм. Така че в началото си използва характеристика на детерминистичен режим което се показва и малката стойност на латентността която се получава в началото докато мрежата не пренатоварена. Като мрежата стане натоварена един начин да е управлява с нея е само чрез използване на адаптивното маршрутиране, за да може да се осъществи това и да се намери алтернативен път в мрежата който не е задръстен или пренатоварен е необходима допълнителна логика. Тогава можем да видим че при пряко натоварване на мрежата можем да получим и малко по висока стойност на латентността.

Добре е да се каже, че при използване на най-подходящ алгоритъм за маршрутиране при положение да използваме топология торус ще стане ако използваме детерминистичен адаптивен маршрутизатор в неговата структура при изграждане на МнЧ суич. Но все пак, ако сме ограничени при налично използване на допълнителна логика, тогава можем да използваме XY алгоритъмът, който ни позволява поне да използваме висока пропускателна способност, но това ще стане за сметка на висока латентност.

Оптимална структура тогава ще рече избиране на топология торус с нейното по малко закъснение и голяма пропускателна способност.

## **4.1 Перспективи**

Перспективите на това изследване може да се много и в различни области на между системни мрежи. Изграждането на високоскоростна комуникация с цел свързване на повече обработващи ядра от типа, показан на фигура 3.2, където казваме че можем да увеличим скоростта на предаване на данни, управлението с потока данни от една страна, гъвкавост която можем да се заложи при прототипизиращото програмируемо устройство (FPGA), например алгоритмът за маршрутиране. Но това което е друга страна на развиване на такива мрежи който се използват при свръх високо скоростни комуникации но който не искаме да са базирани на съществуващи споделени шини с данни или пак комутиране на електрически вериги.

Най новата обработка на тези сигнали по мрежите чрез използване на високо скоростни чипове мрежа за маршрутиране ще рече че е скоростта постигната на това ниво е почти на 1Tb/s така че това показва че при скоростните телекомуникации мора да има и ефективни алгоритми на маршрутиране в една рака а докато в друга ръка е да използваме скоростта на самата топология и нейните предимства.

## **4.2 Приложения и внедряване**

### **4.2.1 Приложения**

Най-вече в скоростните комуникации сървър-сървър или при използване на междусистемни мрежови връзки между сървър и бекхаул (backhaul) модул в телекомуникациите. Използването на комуникация тип бекхаул модул и сървър има най-много приложения в клетъчните GSM комуникации и това за определяне на достатъчно канали, които ще бъдат използвани от мобилните GSM устройства за пренос на данни като 3G, 4G, 5G итн. Също така може да се осъществи междусистемни мрежа между два или повече сървъри.

### **4.2.2 Внедряване на резултатите**

Резултатите от дисертационния труд са внедрени в следния проект:

- Научноизследователски проект в помощ на докторант бр. 161ЦФ002-09 на тема „Изграждане на тест-платформа за оценка на качествените параметри на NoC –(Network on Chip)” с ръководител доц. д-р инж. Валентин С. Моллов

## **НАУЧНО-ПРИЛОЖНИ И ПРИЛОЖНИ ПРИНОСИ**

### **НАУЧНО-ПРИЛОЖНИ**

1. Подробно са изследвани параметрите на съвременните мрежови топологии, като акцентът е върху мрежите, които може да се проектират като Network on Chip (Мрежа на Чип - МнЧ);

2. Предложени са симулатори с подходящи симулационни модели за оценка на производителността, латентността, изразходвана енергия и другите качествени показатели на мрежовите топологии.
3. Избрани са адекватни симулационни сценарии и са приложени с оглед получаване на оптимална оценка на параметрите на МнЧ и за преодоляване на грешки.

## **ПРИЛОЖНИ**

1. Предложени са набор от методи и средства за подобряване на качествено изследване на МнЧ чрез предлагане на подходяща хардуерна тест-платформа.
2. Проектиран и реализиран е модел на смарт Ethernet суич чрез синтезиране код с разширена функционалност и управление на потока. Добавена е разширяваща платка за ускорение потока на данни.

## **СПИСЪК НА ПУБЛИКАЦИИТЕ ПО ДИСЕРТАЦИОННИЯ ТРУД**

### **В рецензирани научни списания**

1. Ile Dimitrievski and Valentin S. Mollov, Evaluation of the routing algorithms with reduced energy consumption. Journal of Computer & Communications Engineering, v.10, No 1/2016, pp.27-31, ISSN 1314-229
2. Ile Dimitrievski and Valentin S. Mollov, Comparison Between Routing Algorithms Applied in NoC Architectures for Smart Ethernet Switches Routing Schemes, Journal of Computer & Communications Engineering, v.3, No 1/2018 (приета, под печат)

### **В международни и национални научни конференции**

3. Ile Dimitrievski, Valentin S. Mollov, Novel abstraction and prototyping methods for NoC- based smart Ethernet switches. Proceedings of the Scientific Works of the Union on Scientist in Bulgaria - Plovdiv, 2015, pp.169-172, ISSN 1311-9419
4. Ile Dimitrievski, Valentin S. Mollov, Performance testing methods for NoC-based smart Ethernet switches. Proceedings of the Scientific Works of the Union on Scientist in Bulgaria - Plovdiv, 2015, pp.173-176, ISSN 1311-9419
5. Ile Dimitrievski, Valentin S. Mollov, Application of advanced routing schemes for energy consumption optimization in NoC-based Ethernet switches. Proceedings of the Advanced Aspects of Theoretical Electrical Engineering Sofia, 2016, pp.193-199, ISSN 1313-9487

6. Ile Dimitrievski, Valentin. S. Mollov, Overview and simulation of NoC - based smart Ethernet switches. Proceedings of the Advanced Aspects of Theoretical Electrical Engineering Sofia, 2016, pp.200-206, ISSN 1313-9487
7. Ile Dimitrievski and Valentin S. Mollov, Comparison Between Routing Algorithms Applied in NoC Architectures for Smart Ethernet Switches Routing Schemes, 8<sup>th</sup> International Scientific Conference Computer Science'2018, 13-15 September, pp.34-42, ISBN: 978-619-167-177-9

## SUMMARY

Nowadays fast communications need to achieve some very special requirements that are based both on the architecture and implementation.

We are now living in the era of interconnection between every single device in the global network the Internet service and his revolution is already expanding achieving some of the fastest speeds ever.

The Internet service requires some of the challenging architecture improvements over the existing carrier protocols and the networks topologies. Solving one of this major problem it's an interconnection switching and delivery of the data from the one of the existing machine to another. We have proposed an improved smart Ethernet switch which has low latency, high throughput, and low-power mode with the dedicated new smart routing protocols.

This switch is practically based on the Network on Chip (NoC) communication network. In the future it's possible to integrate some of the computing nodes on the same chip without need for the external processing power from the dedicated block like Digital Signal Processor (DSP), Floating point unit (FPU), FPGA (Field Programmable Gate Array) and Input Output (IO).

Achieving speeds of almost 1Tbps on such NoC is very important and give as future research direction for such global future communications in the world.