



# 基于FPGA的片上网络自适应路由算法的设计与实现

答辩人：江岑倩 指导老师：蔡旻

# 目录

## CONTENTS

1

选题背景

2

研究意义

3

相关工作

4

我的工作

5

实验与结果

6

结束语

## 选题背景



### 片上网络自适应路由算法

多核芯片上不同核心运行不同类型的应用：数据密集型 vs. 计算密集型  
自适应路由算法比确定性路由算法更能适应不均匀的片上网络流量分布。



### 蚁群优化 (ACO)

源自自然界中的蚁群觅食行为，模拟基于信息素的信息交换与反馈机制  
算法呈现出并行与分布特性。



### FPGA

半定制、效率高、功耗低、是多核体系结构模拟的主要加速技术之一。

## 研究意义

研究片上网络自适应路由算法能有效改善片上网络的性能与功耗，从而提高多核架构的整体运行效率。FPGA 作为一种高效能的硬件仿真方法，能对大规模片上网络的性能与功耗进行仿真。

据此，本课题基于蚁群优化(Ant Colony Optimization, ACO)思想，采用适合于 FPGA 实现的信息素表的更新方法和蚂蚁包的路由选择方法，来设计应用于 NoC 的自适应路由选择算法。本课题工作为蚁群优化思想的实际应用以及 NoC 路由算法的 FPGA 实现提供了重要的参考价值。

## 相关工作



### 自适应路由算法

用途：从所有输出端口列表中选出可用的输出端口列表。

如Odd Even Turn Model(奇偶转弯)路由算法。



### 选择算法

用途：从可用的输出端口列表中选出最合适的输出端口。

如Buffer Level(缓存水平)和Neighbor on Path(NoP, 路径邻居)算法。



### 基于蚁群优化的自适应路由选择算法

基本思想：结合了基于信息素的路由表设计以及基于缓冲水平的本地模型。

如AntNet。

## AntNet算法原理

是一种基于蚁群优化的自适应路由选择算法。

**基本思想：**结合了基于信息素的路由表设计以及基于缓冲水平的本地模型。

**选择概率的计算：** 
$$P'(j, d) = \frac{P(j, d) + \alpha L_i}{1 + \alpha(|N_k| - 1)}$$

$P(j, d)$  为路由表中  $(j, d)$  位置的信息素浓度值； $L_i$  代表了邻居结点的缓冲水平； $N$  为当前结点的邻居结点数； $\alpha$  为路由选择因子； $P'(j, d)$  即为发往目的地结点  $d$  的消息包、会从  $j$  输出端口输出概率。

**信息素浓度的更新：**

情形1：结点在蚂蚁包的memory中时， 
$$P(i) = P(i) + r \times (1 - P(i))$$

情形2：结点不在蚂蚁包的memory中时， 
$$P(i) = P(i) - r \times P(i)$$

$P(i)$  为信息素表浓度值， $r$  为激励因子。

## FPGA设计与实现

在已有的NoC FPGA实现代码基础上（已实现了XY路由算法）：

实现了**奇偶转弯路由算法**。

实现了**三种Synthetic Traffic**：Uniform，Transpose和Hotspot。

新增两种消息包：**正向蚂蚁包和逆向蚂蚁包**。

采用了**基于信息素的路由表结构**。

采用了**结合了信息素浓度和邻居结点缓冲水平的路由选择机制**。

**对蚂蚁包的处理：**

- 正向蚂蚁包和数据包采用路由表进行路由；
- 正向蚂蚁包到达目的结点后转换为逆向蚂蚁包，并在原路返回的过程中更新所经结点路由表中的信息素浓度。

# 基于FPGA的片上网络自适应路由算法的设计与实现

## 消息包结构

类型	名称	描述
logic	ant	标记消息包的类型 1: ant packet; 0: normal packet
logic [\$clog2(`X_NODES)-1:0]	src_x(y)	该消息的源结点坐标
logic [\$clog2(`X_NODES)-1:0]	dest_x(y)	该消息的目的地结点坐标
logic	backward	标记蚂蚁包的类型（仅为蚂蚁包时有效） 1: backward packet; 0: forward packet
logic [\$clog2(`NODES)-1:0]	num_memories	memory中记录的路径的结点数 （仅为蚂蚁包时有效）
logic [0:`NODES-1][\$clog2(`X_NODES)-1:0]	memory_x(y)	蚂蚁包走过的路径队列 （仅为蚂蚁包时有效）



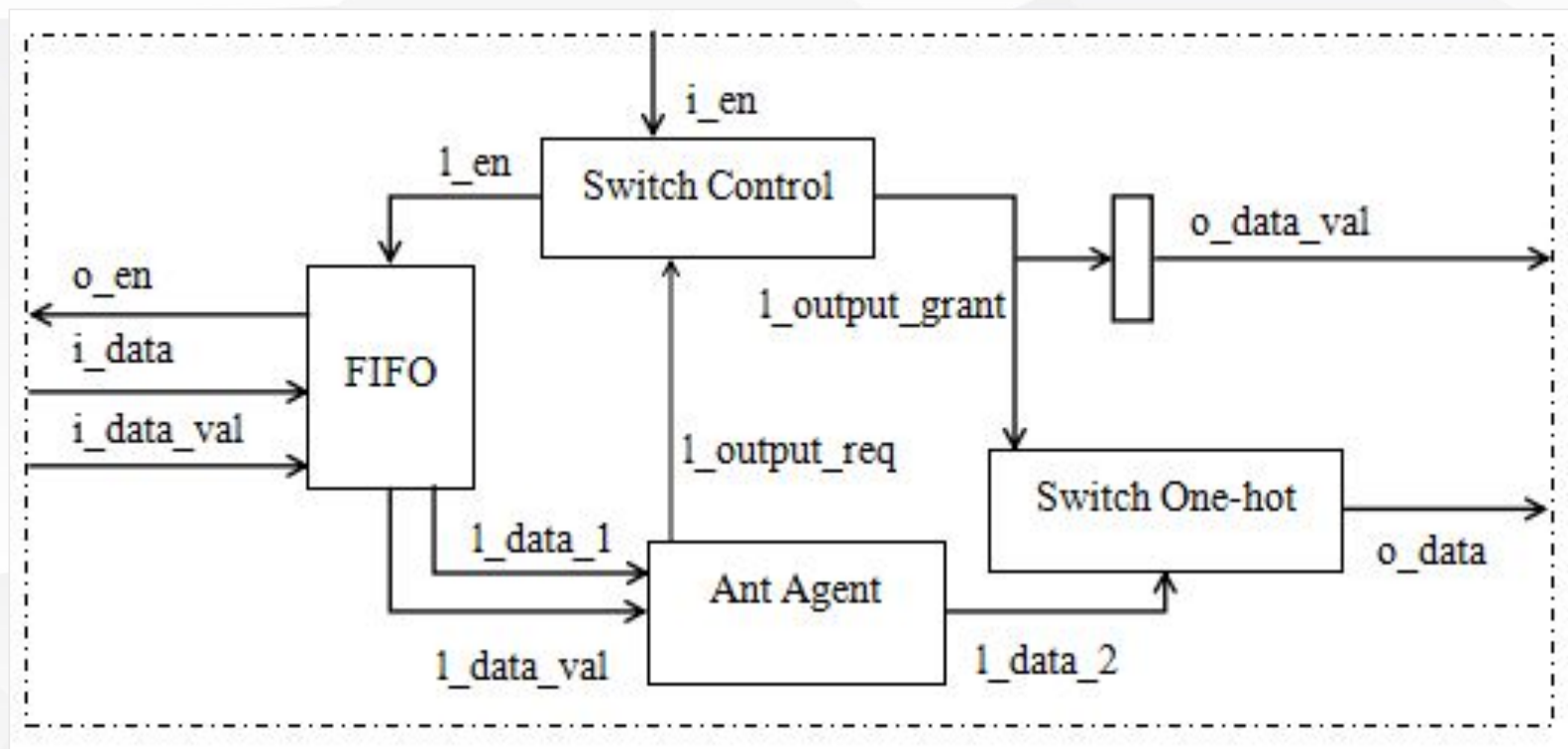
# 基于FPGA的片上网络自适应路由算法的设计与实现

## 信息素表结构

目的结点\输出端口	北	东	南	西
0				
1				
:				
<b>d</b>	ph[d][北]	ph[d][东]	ph[d][南]	ph[d][西]
:				
结点数-2				
结点数-1				

# 基于FPGA的片上网络自适应路由算法的设计与实现

## 路由器架构



## 实验环境

**开发环境**：Altera Quartus Prime 15.1

**硬件描述语言**：SystemVerilog

**开发流程**：FPGA设计收入->仿真->输出实验结果->实验结果统计

- 利用testbench对单个结点和整个网络进行测试
- 使用\$display语句将片上网络及所测算法的性能指标结果输出到文件中

## 实验配置

**拓扑**：网格 ( Mesh ) 拓扑结构， $4 \times 4 = 16$ 个结点

**缓冲长度**：4 (单位：包)

**Synthetic Traffic**：Uniform , Transpose和Hotspot

**包注入率** ( Packet Injection Rate )：0.01 ~ 0.6(单位：包/结点/时钟周期)

**模拟的最大时钟周期数**(Max Cycles)：

预热: 1000, 详细模拟: 10000, 准备排空: 3000, 排空: 3000

**ACO路由选择算法的参数**：

选择因子 $\alpha=0$ (暂未考虑缓冲水平)；激励因子 $rf=1$ （采用 $n$ 位计数器实现）

## 性能指标

**吞吐率** ( Throughput ) = #完成传输的包数量/结点/时钟周期。

**平均包延时** ( Average Packet Delay ) 。

**消息包结构体中用来辅助测试的字段：**

类型	名称	描述
Logic [7:0]	id	仿真时记录消息包的id号
logic	measure	仿真时记录测试状态
logic [`TIME_STAMP_SIZE-1:0]	timestamp	仿真时记录数据包进入网络时的时间

## 实验环境

**开发环境**：Altera Quartus Prime 15.1。

**硬件描述语言**：SystemVerilog。

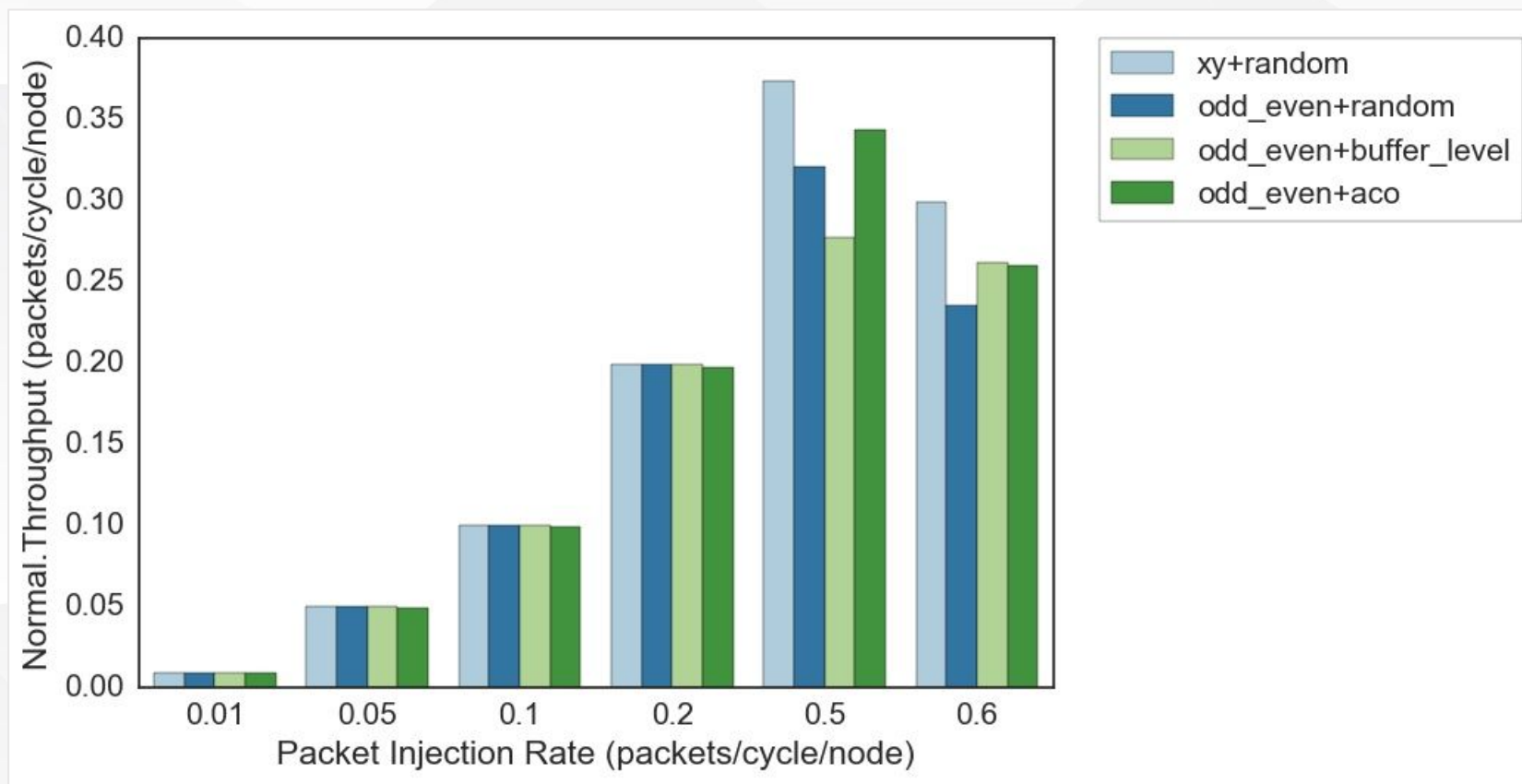
**开发流程**：FPGA设计收入->仿真->输出实验结果->实验结果统计。

- 利用testbench对单个结点和整个网络进行测试；
- 使用\$display语句将片上网络及所测算法的性能指标结果输出到文件中。

# 基于FPGA的片上网络自适应路由算法的设计与实现

## 实验结果：吞吐量

(a) Uniform输入下的吞吐量

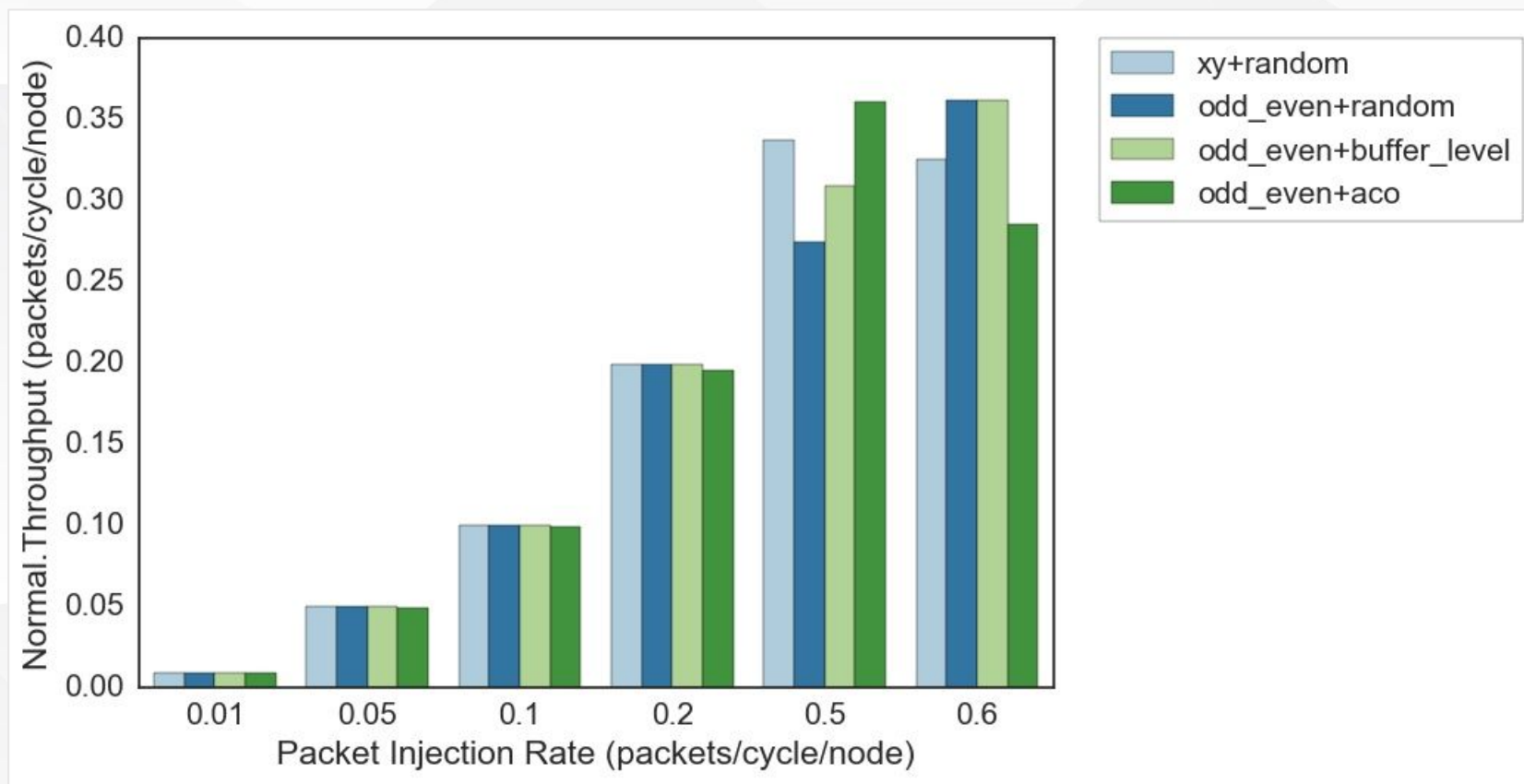


在消息包注入率较低时，差别不明显。在消息包注入率较高时，时好时坏。

# 基于FPGA的片上网络自适应路由算法的设计与实现

## 实验结果：吞吐量

(b) Transpose输入下的吞吐量



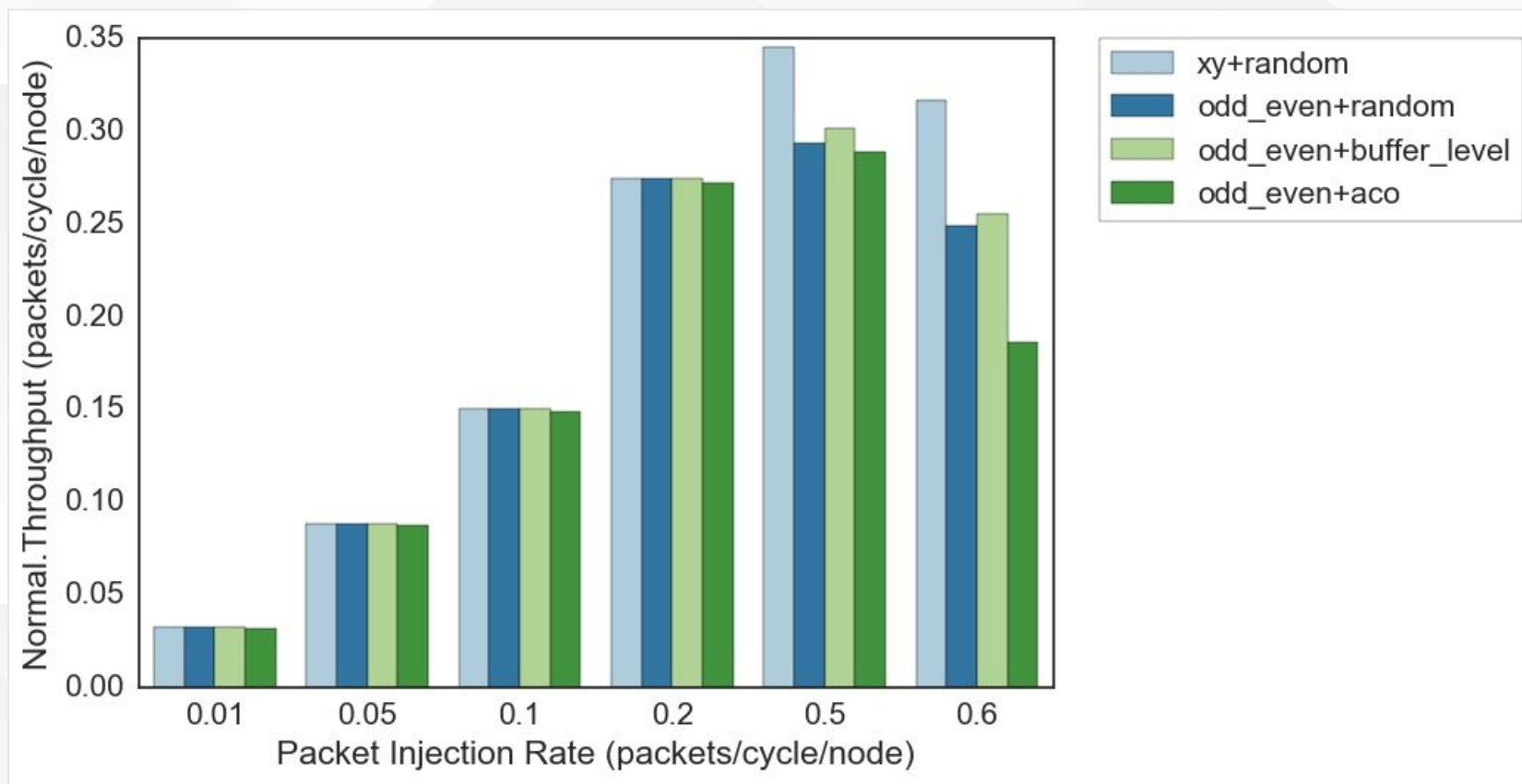
在消息包注入率较低时，差别不明显。在消息包注入率较高时，时好时坏。



# 基于FPGA的片上网络自适应路由算法的设计与实现

## 实验结果：吞吐量

(c) Hotspot输入下的吞吐量

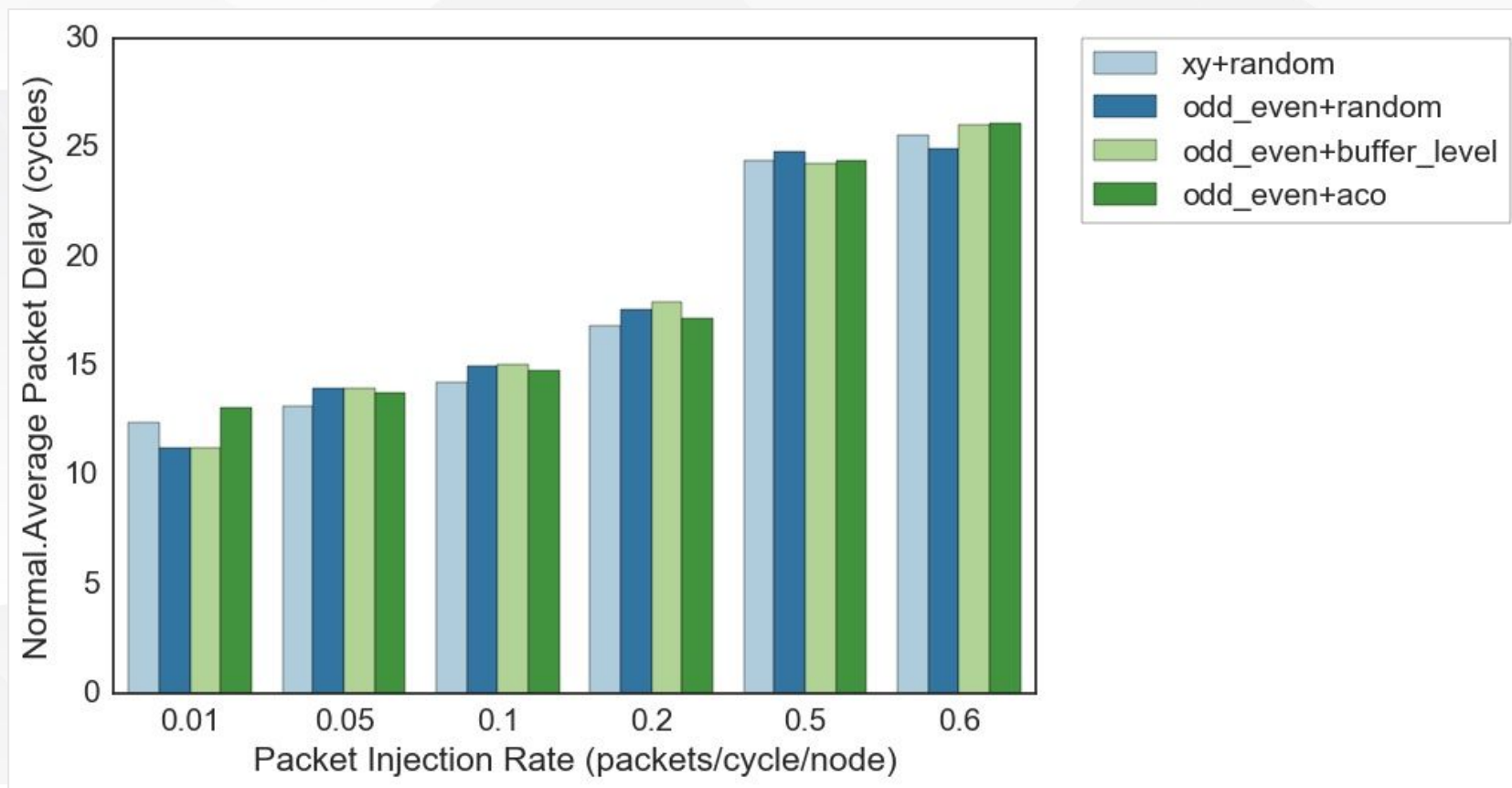


在消息包注入率较低时，差别不明显。在消息包注入率较高时，表现较差。

# 基于FPGA的片上网络自适应路由算法的设计与实现

## 实验结果：包时延

(a) Uniform输入下的平均包时延

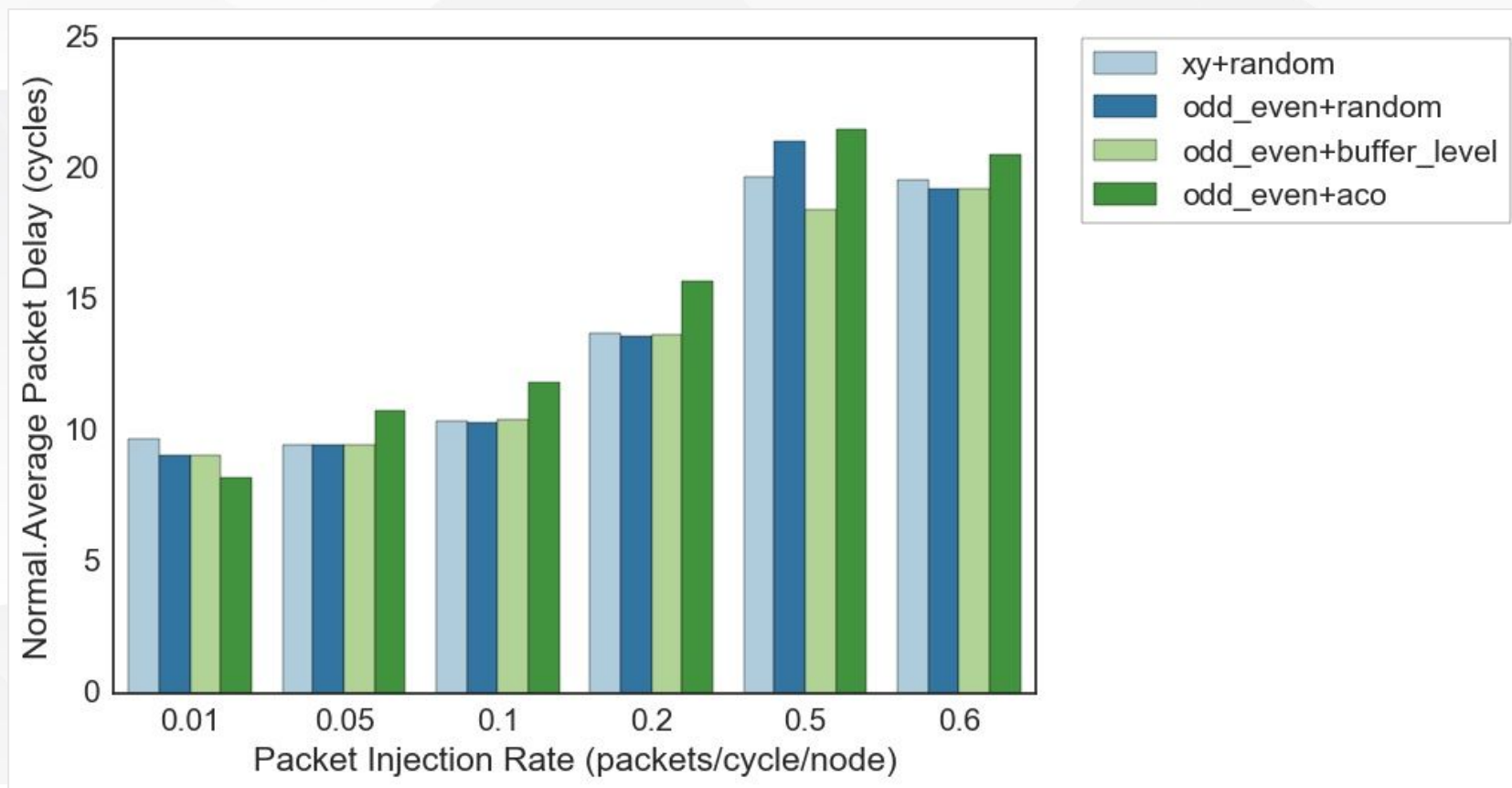


与Buffer Level算法相比，存在少许的差距。

# 基于FPGA的片上网络自适应路由算法的设计与实现

## 实验结果：包时延

(b) Transpose输入下的平均包时延

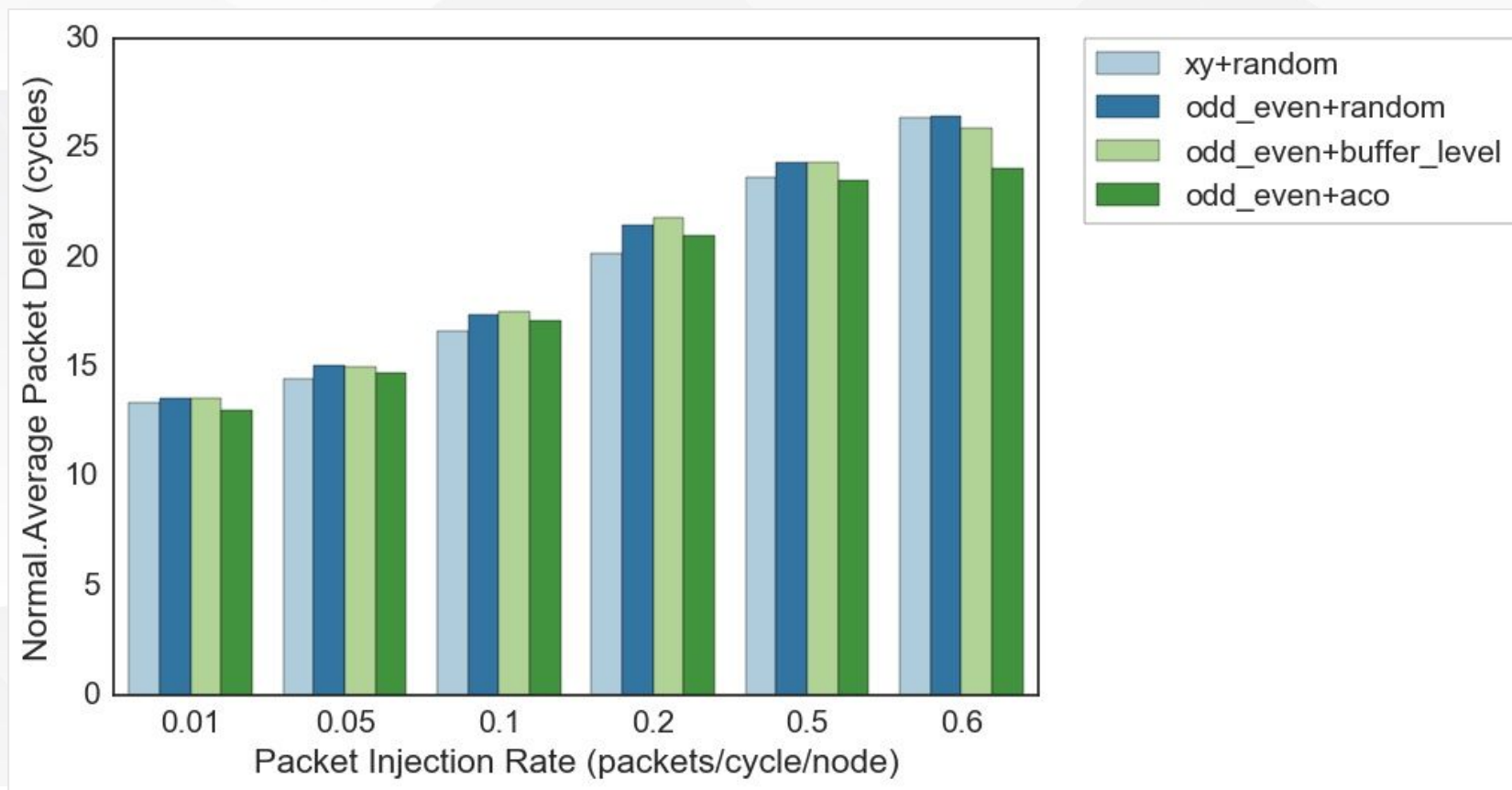


与Buffer Level算法相比，表现得较差。

# 基于FPGA的片上网络自适应路由算法的设计与实现

## 实验结果：包时延

(c) Hotspot输入下的平均包时延



与Buffer Level算法相比，表现出一些优势。

## 结束语

### **FPGA仿真实验结果小结：**

相较于Buffer Level选择算法，文中实现的ACO选择算法在Uniform、Transpose、Hotspot三种输入模式下：

- 能达到的吞吐率最大提升幅度为+23.81% (PIR=0.5)、+16.69% (PIR=0.5) 和-0.96% (PIR=0.01)；
- 能达到的平均包时延最大降低幅度为+3.90% (PIR=0.2)、+9.73% (PIR=0.01) 和+7.18% (PIR=0.6)。

### **工作中不足之处与可能的改进：**

- 实验中采用4x4 Mesh结构，结点数偏少；
- 在FPGA中实现的NoC架构不是很完善，如未考虑链路传输时延等；
- 在进行选择时未加入网络当前状态的考虑，且未根据信息素表浓度值进行概率选择，而是直接选择最大值对应的下一跳结点；
- 蚂蚁包与数据包的结合、蚂蚁包的注入率问题以及每个参数的设置都会影响到实验结果。



# 谢谢各位老师批评指正

北京工业大学计算机学院本科毕业设计答辩