

Отчет по проделанной домашней работе «TokenRing»

Красильников Григорий Олегович

26 ноября 2017

Целью данной работы является изучение поведения алгоритма Token Ring в зависимости от количества узлов N и количества пакетов P ($1 \dots N$), находящихся в транзите одновременно.

Основные метрики, используемые в данном исследовании: пропускная способность сети (throughput) и характерное время задержки (latency). Для увеличения точности измерений эксперименты проводятся n раз, используемое $n = 5$.

Также предлагается подумать над возможной оптимизацией данного алгоритма.

1. Пусть $N = 20$. Рассмотрим, как меняются вышеназванные характеристики для $P = (2..20)$ с шагом 2.

График зависимости задержки в очереди от P :

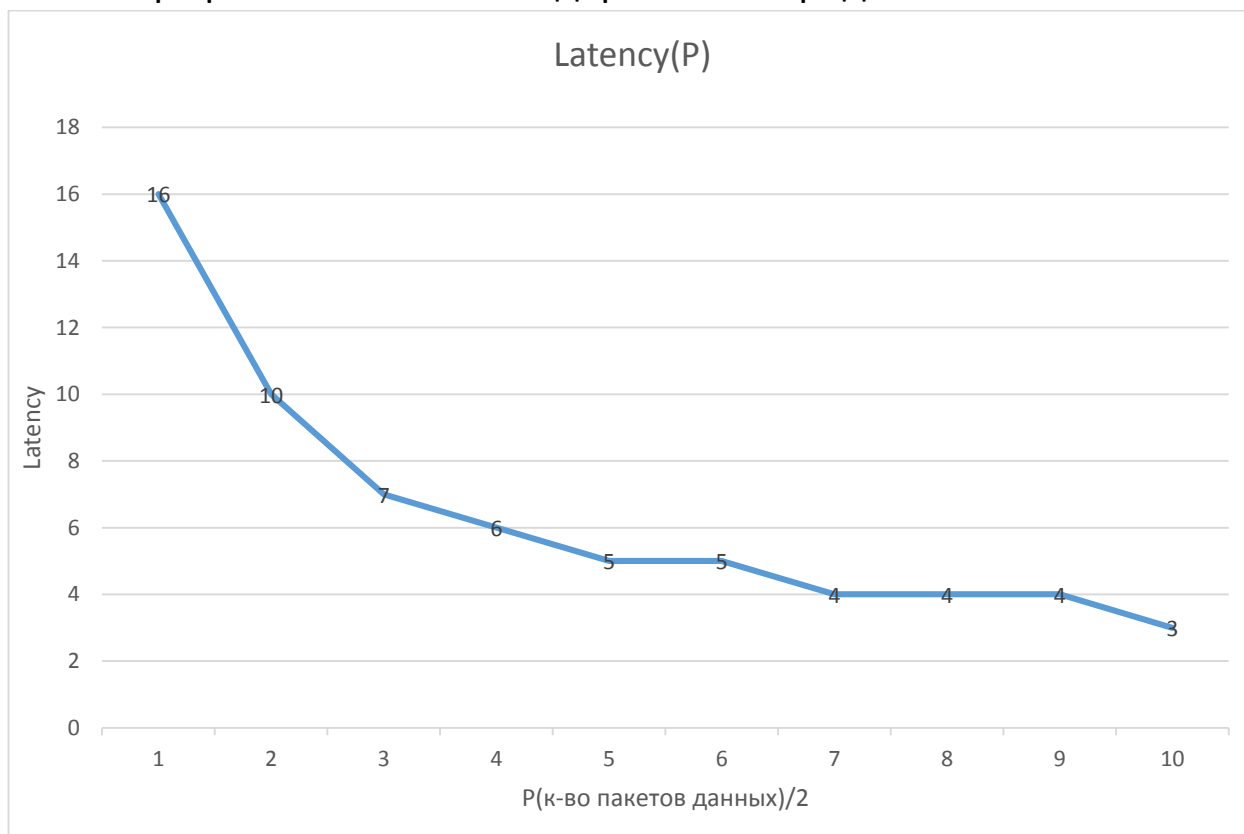


График зависимости задержки доставки от P:

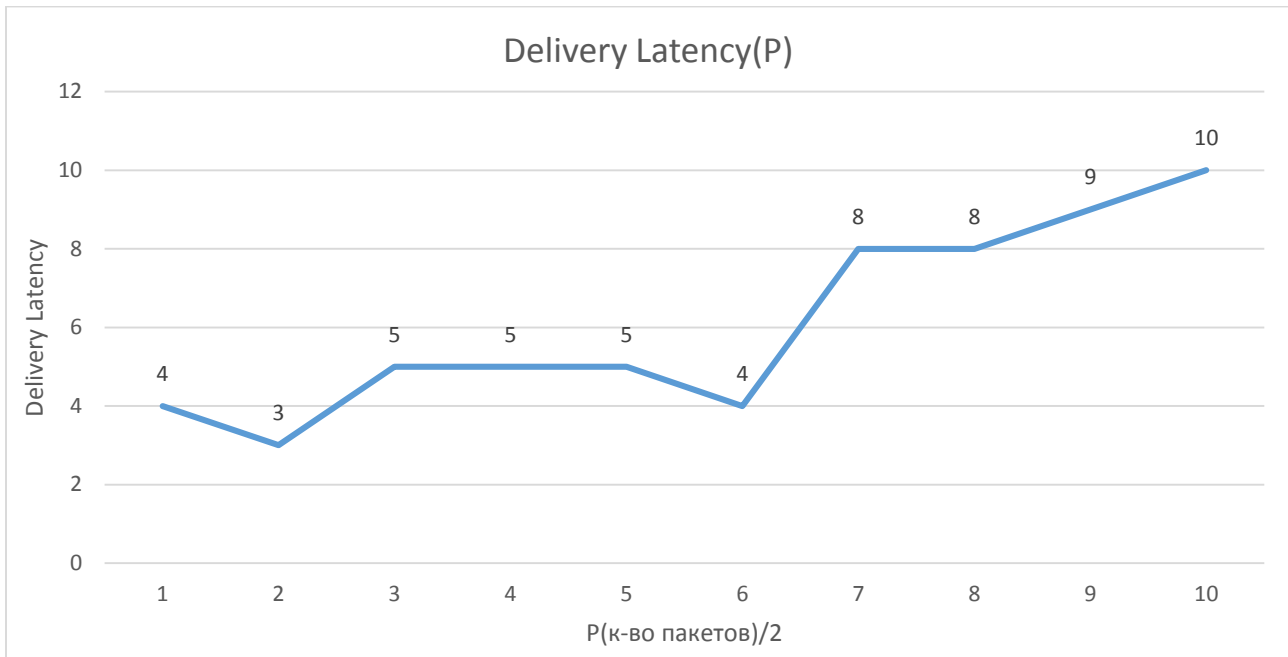
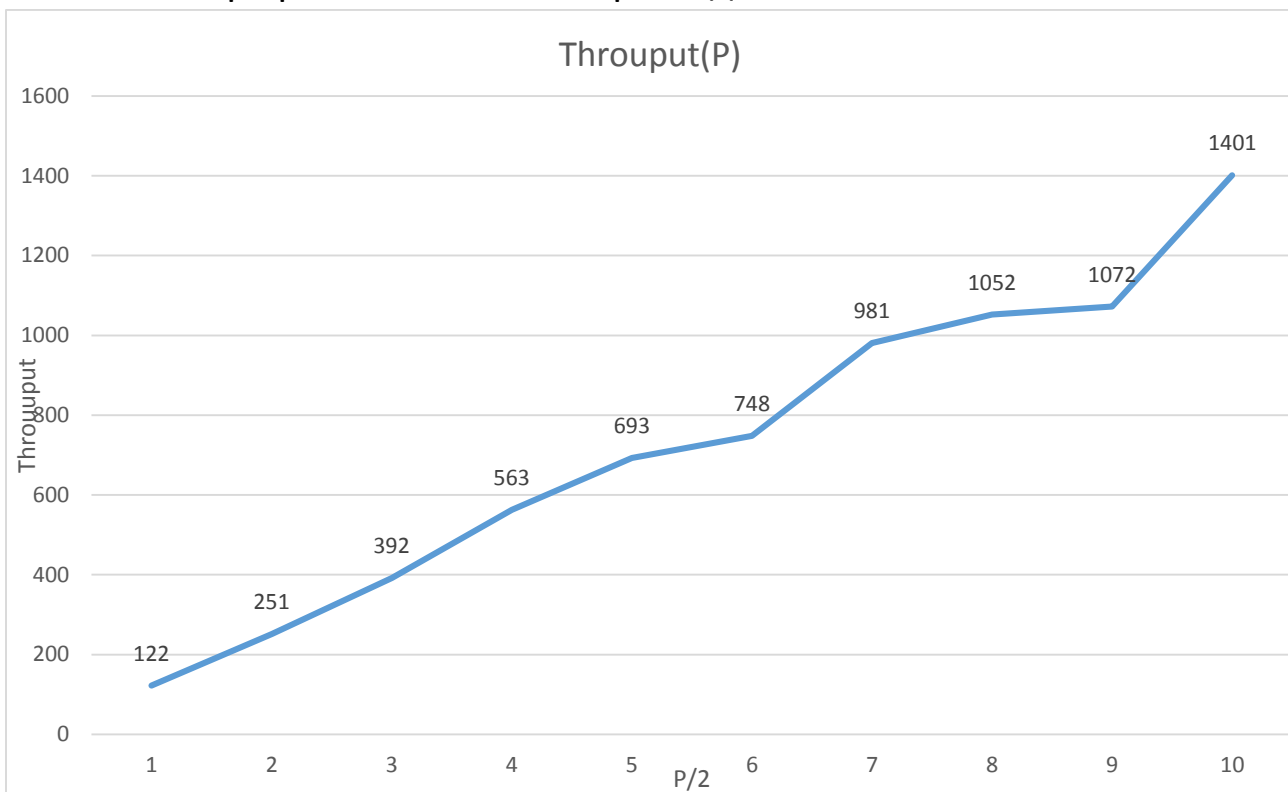


График зависимости прошедших пакетов от P:



Как видно из графиков, время задержки уменьшается с увеличением количества пакетов. Это связано с тем, что количество пакетов в данном эксперименте не превышает количества нод(20). С другой стороны, задержка доставки увеличивается – это можно объяснить тем, что с

увеличением количества пакетов все чаще случается ситуация, когда следующие текущие пакеты ждут отправления предыдущих с ноды. Количество прошедших пакетов возрастает, что достаточно очевидно.

2. Теперь рассмотрим обратную ситуацию – зафиксируем количество пакетов $P=20$ и будем изменять количество нод с 4 до 40 с шагом, равным 4.

График зависимости задержки в очереди от N :

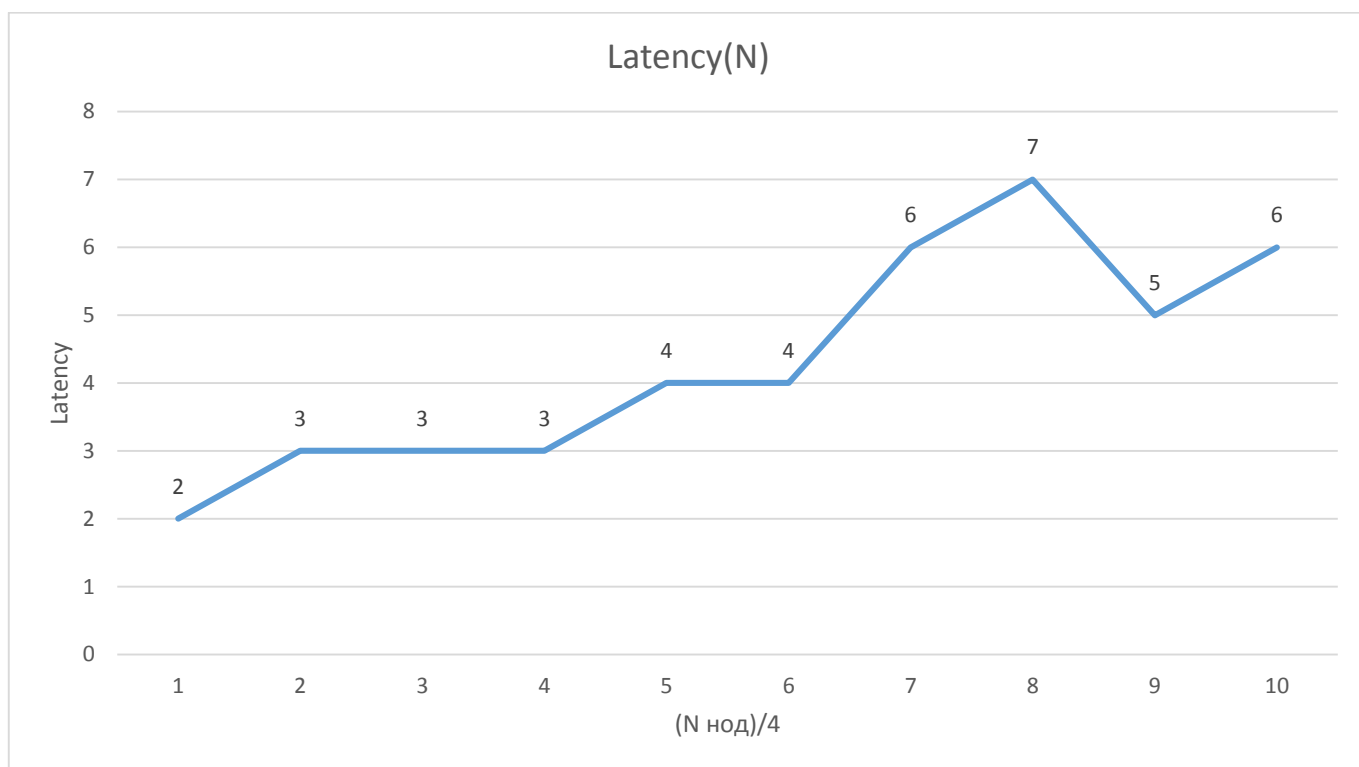


График зависимости задержки доставки от N :

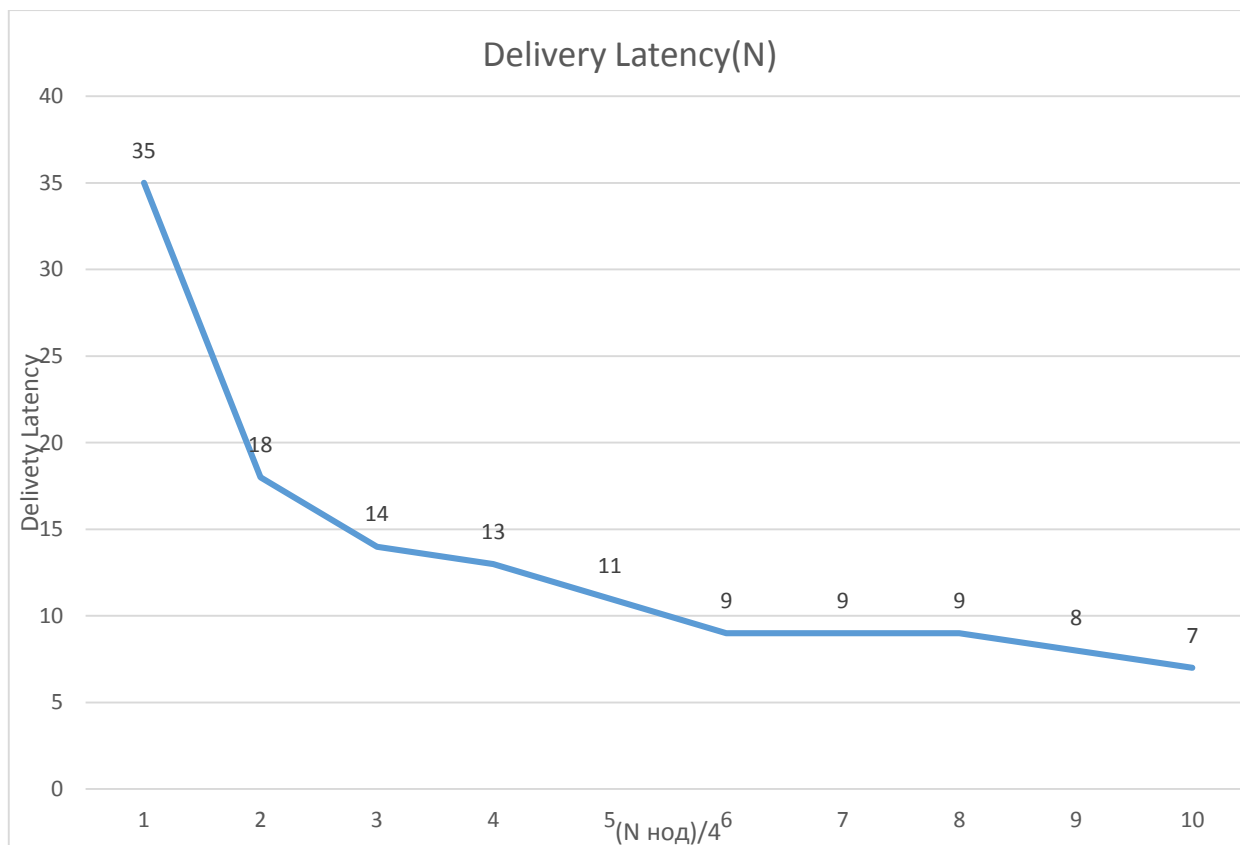
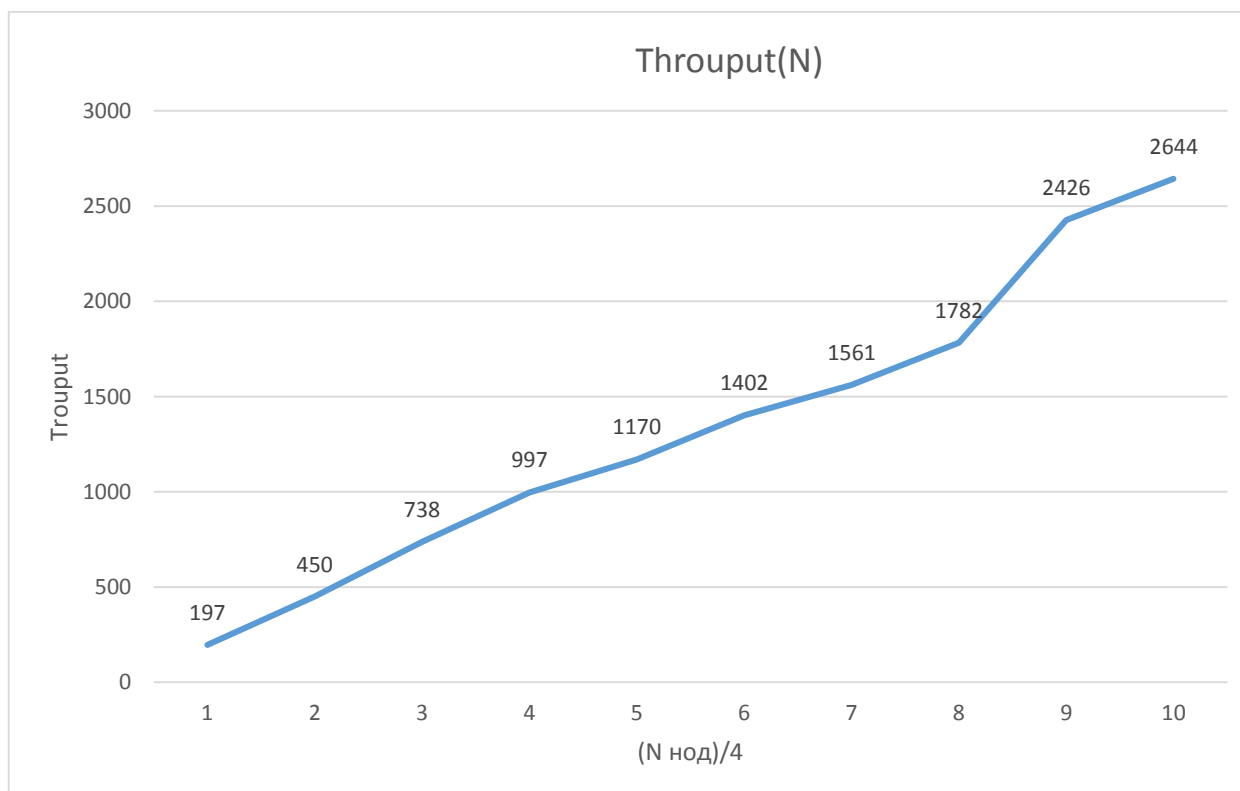


График зависимости прошедших пакетов от N:



Из этих графиков мы можем сделать следующие выводы: задержка в lock'е изменяется не сильно и больше похожа на случайные флуктуации, задержка доставки падает при увеличении количества нод, что логично, потому что мы зафиксировали к-во пакетов – 20. В начале, когда пакетов сильно больше, чем нод, задержка довольно большая, но позже резко падает. Количество пакетов, очевидно, увеличивается при увеличении количества нод.

Интересно, что при попытке автоматизировать эксперименты путем добавления простого цикла(для конкретной пары значений N и P), время задержек снижается в несколько раз в зависимости от исходных данных. Например при N = 20 и P = 10 получились такие результаты:

N=20		P=10		N=20		P=10	
	4		7		5		8
	5		4		4		6
	6		7		3		5
	6		7		2		4
	6		8		2		3

Первая таблица – независимые запуски тестов без цикла, колонки обозначают Latency in Lock и Delivery Latency соответственно. Вторая таблица – результаты работы цикла. Видимо, это связано с оптимизацией самой java.