

Санкт-Петербургский государственный электротехнический университет  
им.

В.И. Ульянова (Ленина)

# Реализация алгоритма сетевого планировщика hClock на базе компонентов с открытым исходным кодом

**Выполнила:** Кочнева Ольга Романовна, гр. 5303

**Руководитель:** Кринкин Кирилл Владимирович, к.т.н., зав. каф. МО ЭВМ

**Консультант:** Соснин Владимир Валерьевич, к.т.н., инженер-исследователь в ООО “Техкомпания Хуавэй”

Санкт-Петербург

2021

# Актуальность

Ни один из существующих сетевых планировщиков не удовлетворяет всем этим свойствам одновременно:

- Открытый доступ.
- Простота интеграции в ОС Linux.
- Поддержка иерархических конфигураций.
- Гарантия минимальной пропускной способности
- Взвешенное перераспределение трафика.
- Ограничение пропускной способности сверху.
- Наличие контроля доступа.

# Информационная база исследования

**Данное исследование основывается на работе:**

*Billaud J. P., Gulati A. hClock: Hierarchical QoS for packet scheduling in a hypervisor //Proceedings of the 8th ACM European Conference on Computer Systems. – 2013. – С. 309-322.*

**Краткое содержание:**

Представлен алгоритм сетевого планировщика под названием hClock, встроенный в гипервизор VMware.

Алгоритм удовлетворяет большинству вышеупомянутых свойств, кроме простоты интегрирования в ОС и доступности.

# Краткая характеристика исследования

**Цель** - снизить сложность и стоимость внедрения hClock-подобного алгоритма управления пропускной способностью в сети за счет применения компонентов с открытым исходным кодом в Linux.

**Объект исследования** - сетевые планировщики.

**Предмет исследования** - способ реализации hClock-подобного алгоритма для планирования трафика сети с использованием компонентов с открытым исходным кодом.

# Задачи

1. Выполнить обзор существующих сетевых планировщиков.
2. Разработать алгоритм, имитирующий алгоритм hClock.
3. Провести эксперименты для анализа разработанного алгоритма.
4. Сформулировать рекомендации по применению разработанного алгоритма.

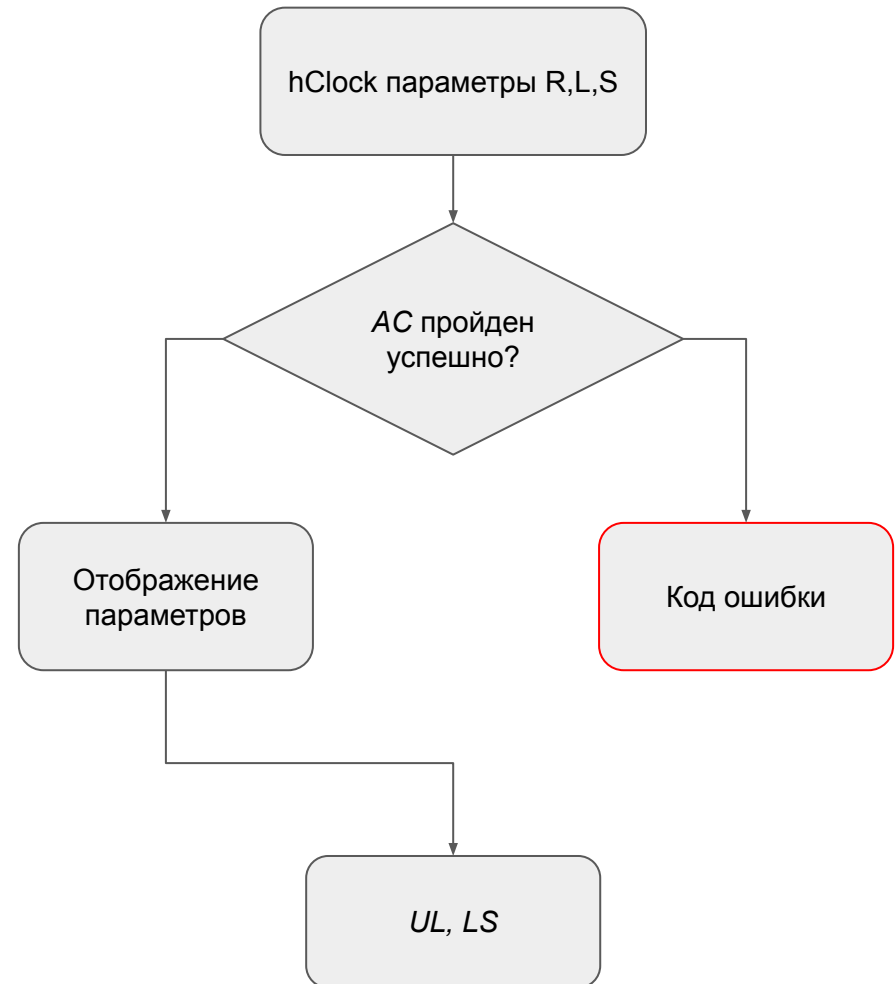
# Обзор

Критерий	SFQ	DRR, QFQ	HTB	HFSC	WFQ, CBWFQ	hClock	BW	SG-QoS
Доступность	+	+	+	+	+	-	-	-
Простота интеграции	+	+	+	+	-	-	-	-
Иерархия	-	+	+	+	-	+	+	+
Резервирование	-	-	-	-	-	+	+	-
Вес	-	+	+	+	+	+	+	+
Ограничение	-	-	+	+	-	+	+	+
Контроль доступа	-	-	-	-	-	+	+	-

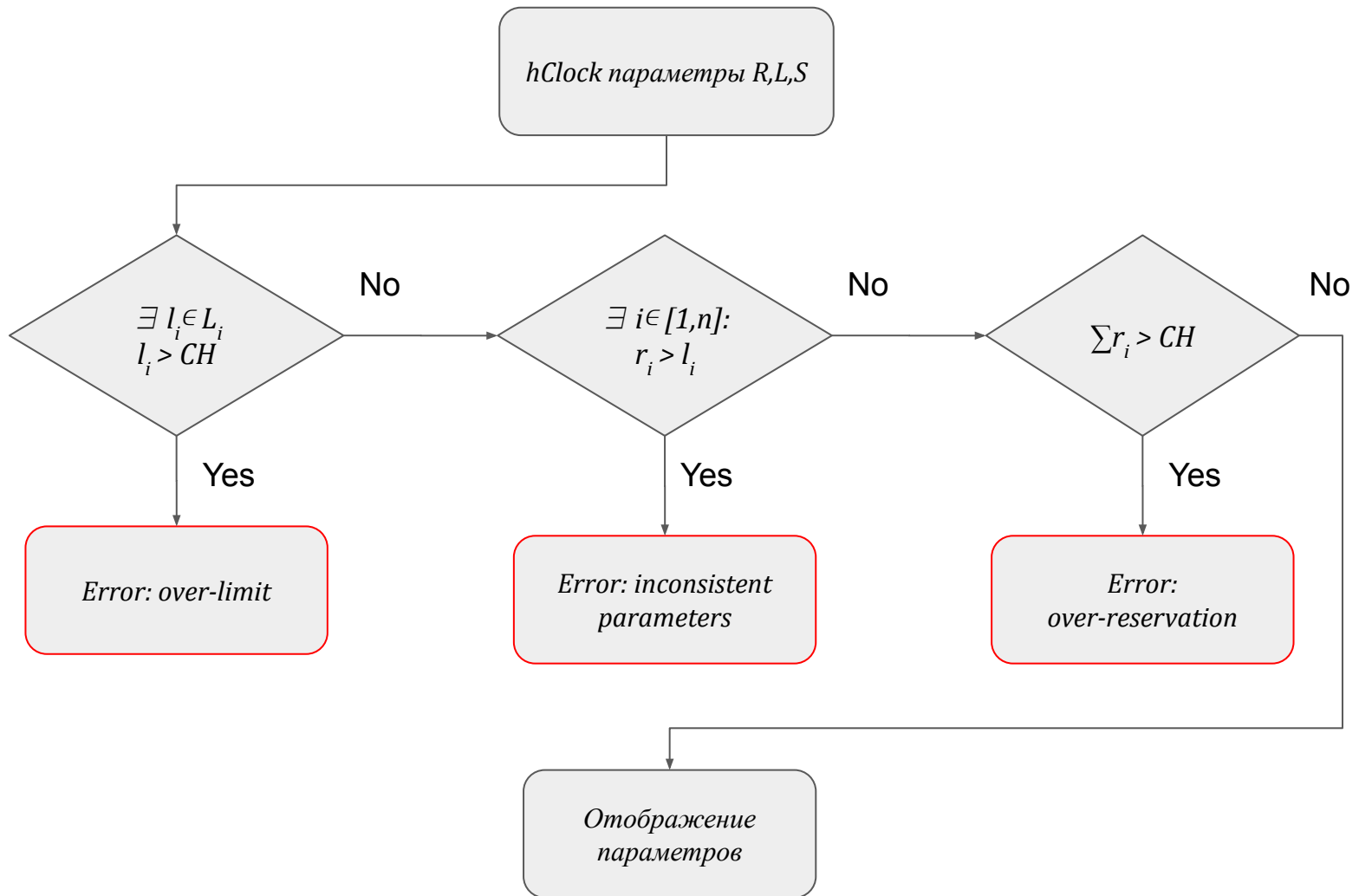
# Алгоритм: блок-схема

Сокращения:

- $n$  = количество классов
- $CN$  = пропускная способность
- $UT$  = нераспределенная пропускная способность
- $AC$  = контроль допуска
- $R = \{r_i\}_{i \in [1, n]}$  - reservations
- $L = \{l_i\}_{i \in [1, n]}$  - limit
- $S = \{s_i\}_{i \in [1, n]}$  - shares
- $UL = \{ul_i\}_{i \in [1, n]}$  - Upper Limit
- $LS = \{ls_i\}_{i \in [1, n]}$  - Link Shares

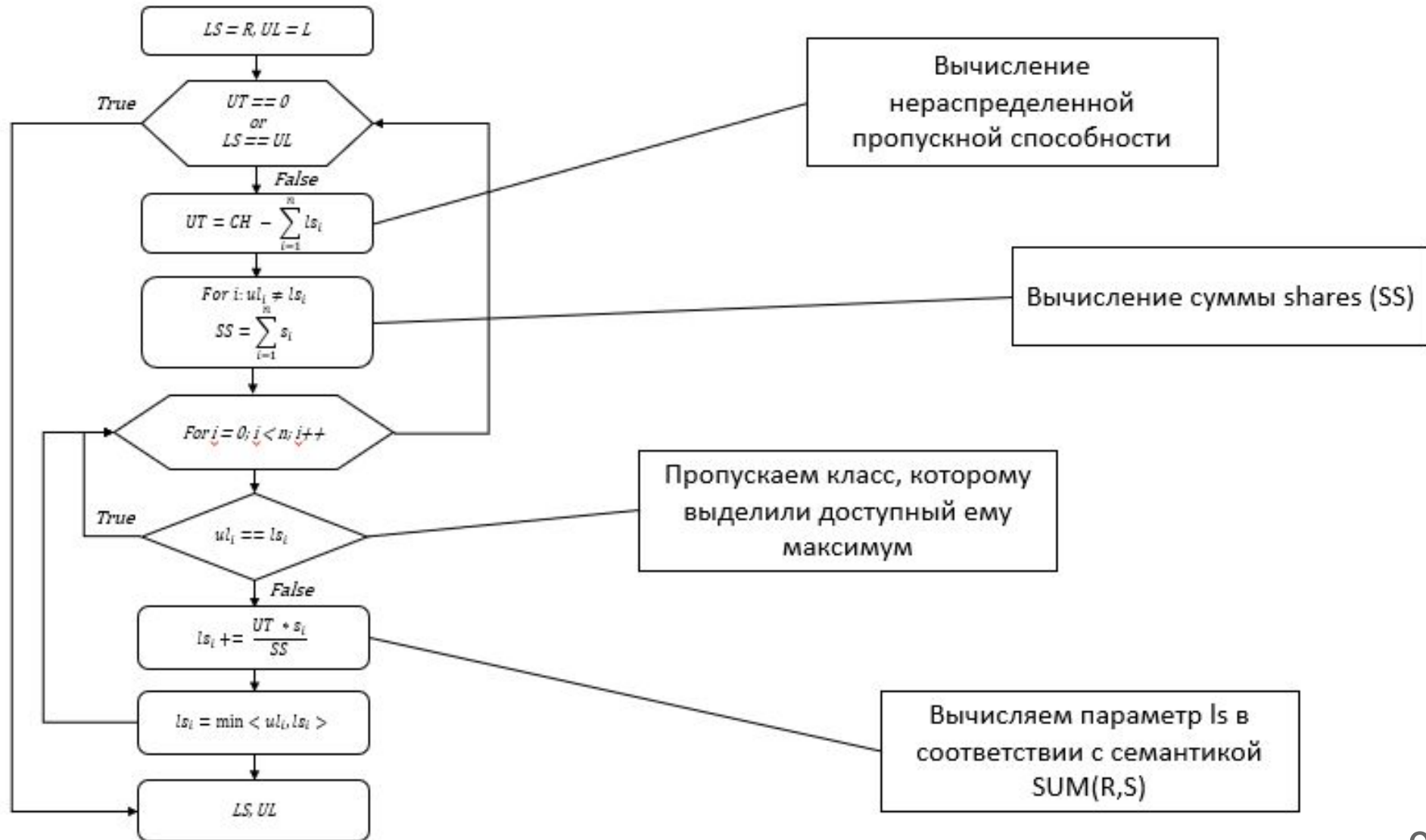


# Алгоритм: контроль допуска в сеть (Admission Control)





# Алгоритм: отображение параметров



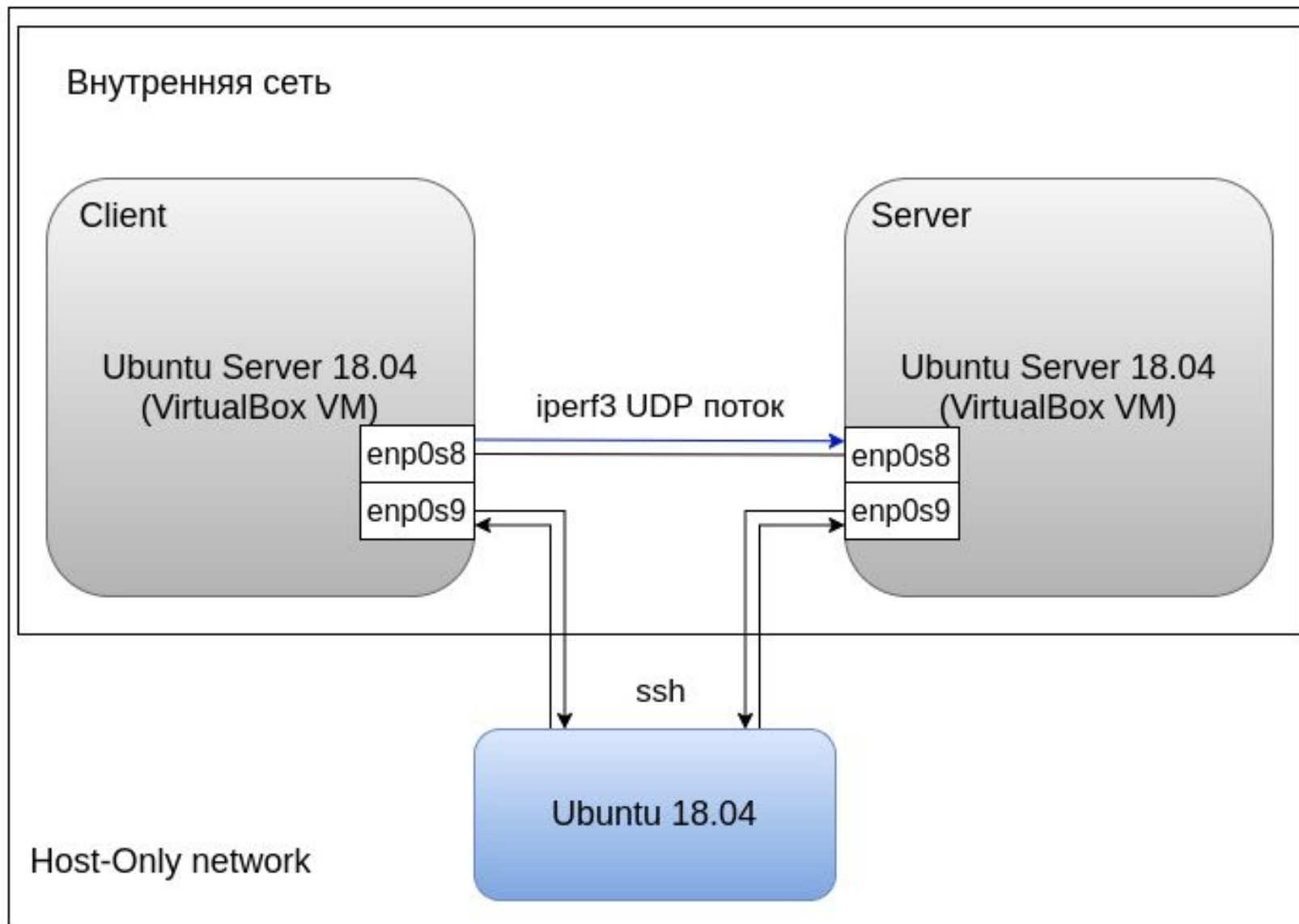
# Алгоритм: SUM(R,S) семантика

Неиспользованная пропускная способность распределяется среди потоков пропорционально значению Shares каждого потока.

**Пример:**  $LS = 10 + 70 * 2 / (2 + 8) = 24$ , где неиспользованная пропускная способность  $70 = 100 - 10 - 20$ .

№ потока	R	L	S	ls/rate	ul/ceil
1	10	100	2	24	100
2	20	100	8	76	100

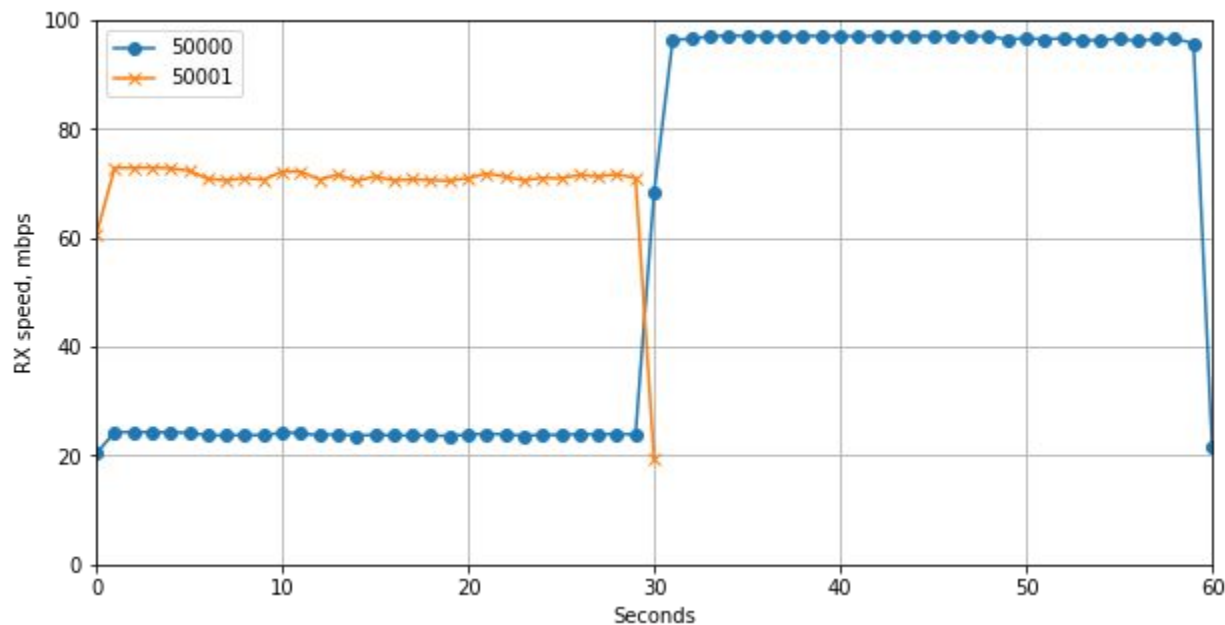
# Тестирование: условия эксперимента



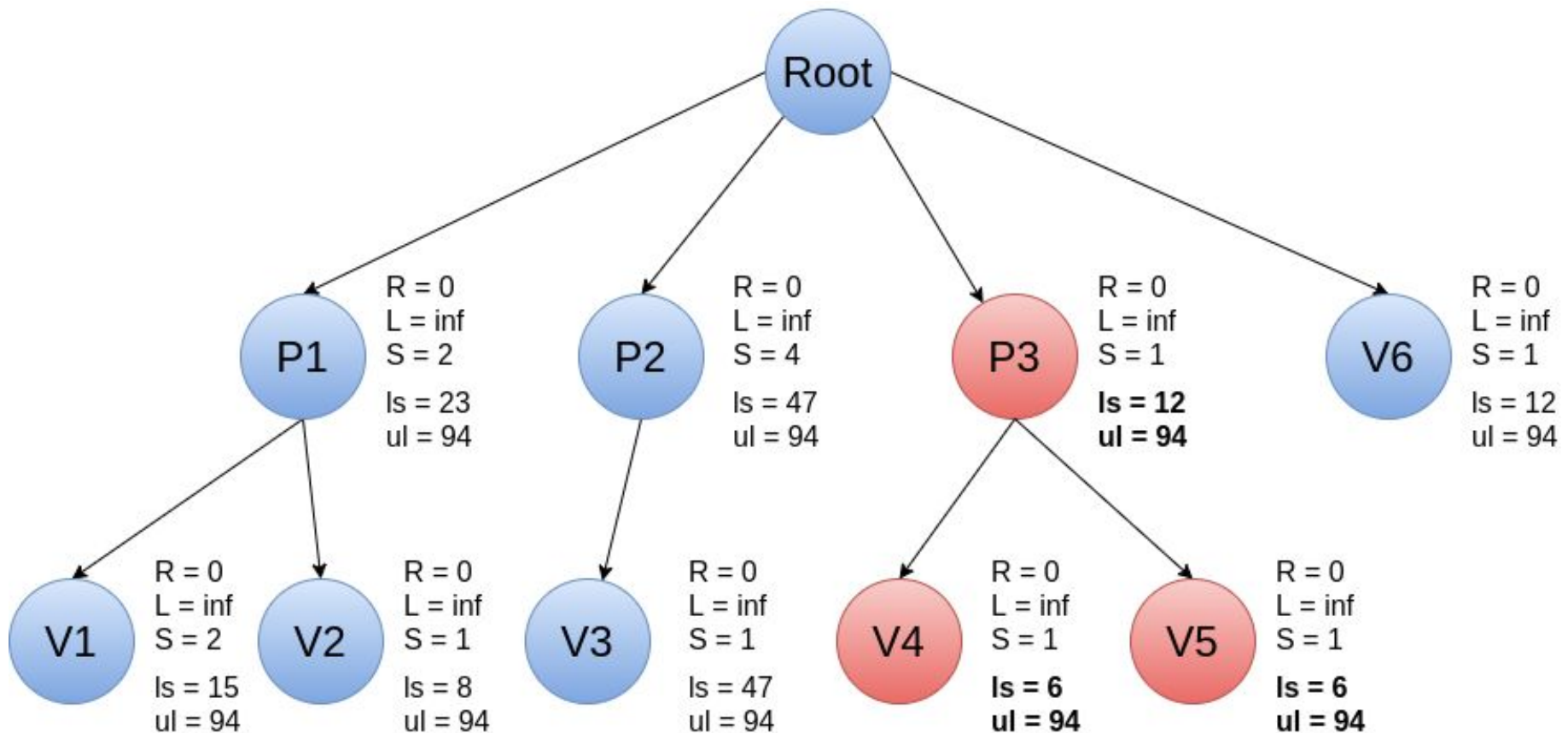
# Тестирование: простая конфигурация

№ потока	R	L	S	ls/rate	ul/ceil
50000	10	100	2	24	100
50001	20	100	8	76	100

Bandwidth sharing experiment №1618303847

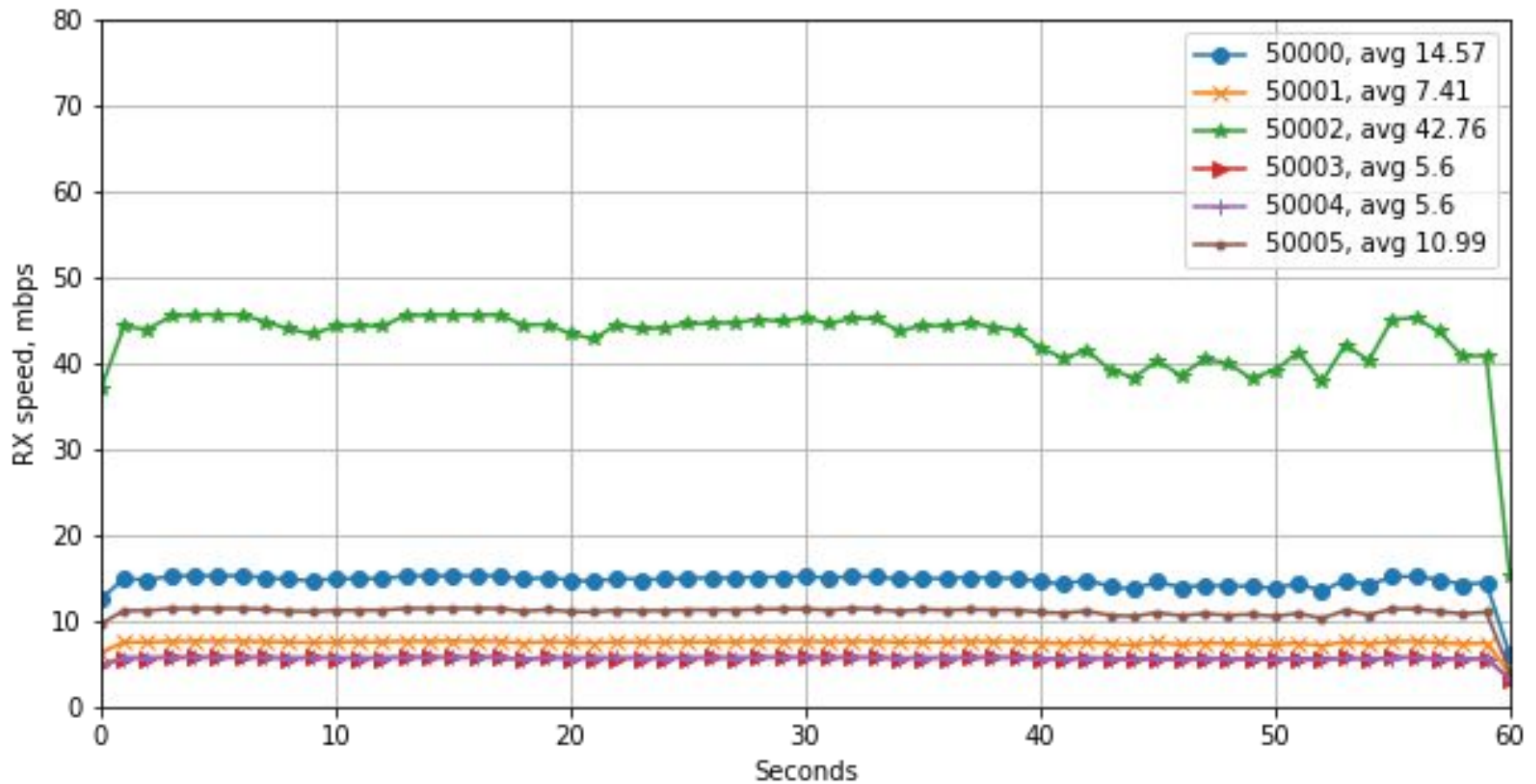


# Тестирование: сложная конфигурация (прим. из статьи hClock)



# Тестирование: сложная конфигурация

Bandwidth sharing experiment №1618328662



# Апробация и рекомендации по применению

1. Выступление на конгрессе “КМУ - X Конгресс Молодых Ученых”.
2. Публикация научной статьи “Алгоритм сетевого планировщика на базе компонентов с открытым исходным кодом” в сборнике трудов конгресса (принята к публикации).
3. Репозиторий проекта: [rls-network-schedulers](https://github.com/rls-network-schedulers)

## Рекомендации по применению

Реализованный алгоритм может быть применим в:

- гипервизорах с открытым исходным кодом
- программных маршрутизаторах
- Android-устройствах и любых других устройствах, ОС которых основана на ядре Linux

# Выводы

Был реализован алгоритм предоставляющий возможность распределять пропускную способность сети обладающий следующими свойствами:

- распределение пропускной способности аналогично распределению, представленному в алгоритме hClock, если все классифицированные потоки насыщенные или при отсутствии резервирования;
- простота интеграции в ОС Linux, не требующая перекомпиляции или реконфигурирования ядра;
- наличие GPL-лицензии.



**Спасибо за внимание!**