

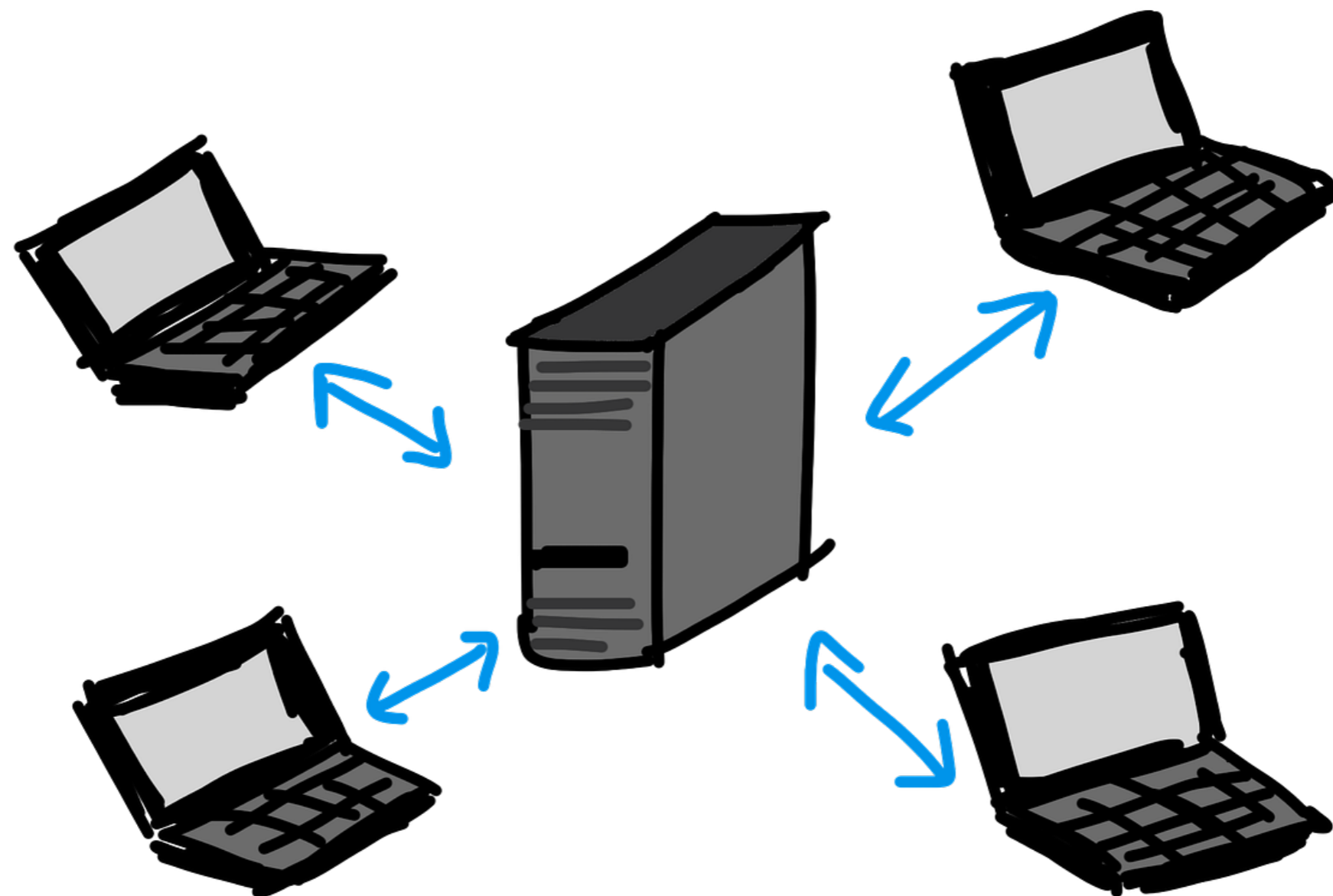
RIPE

Сравнение методов выравнивания трафика

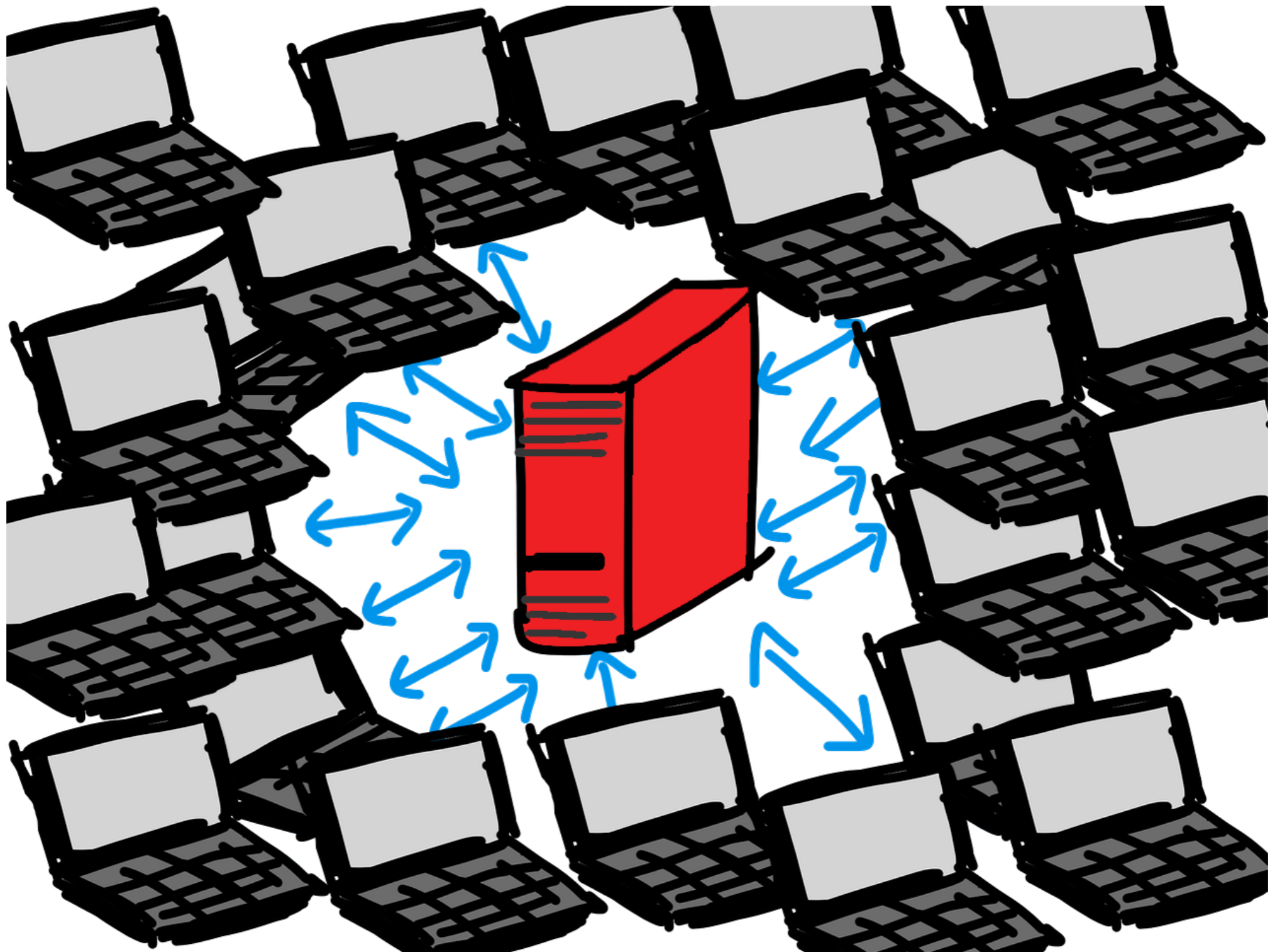
Сергей Зубов



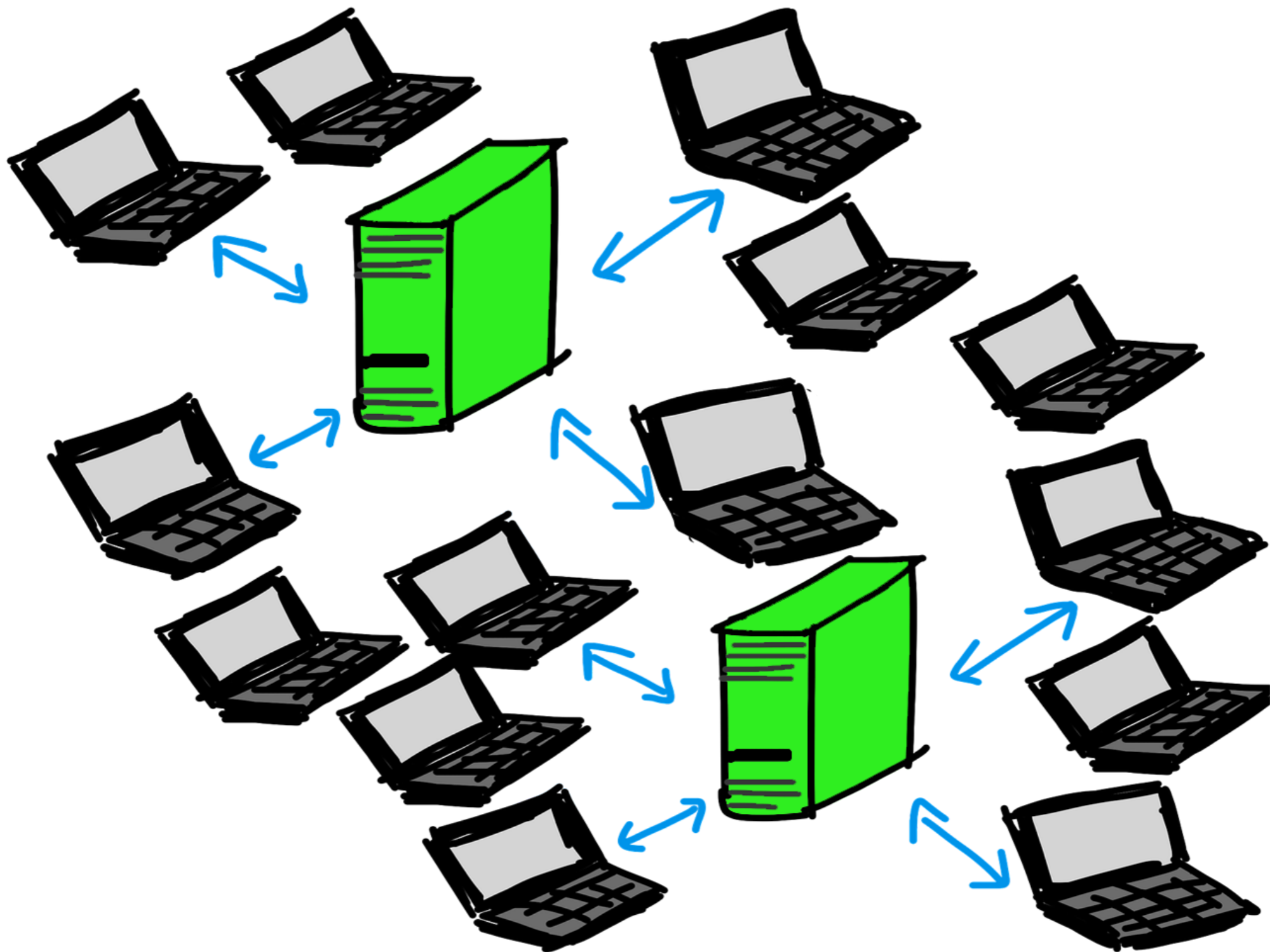
Зачем нужна балансировка?



Зачем нужна балансировка?



Зачем нужна балансировка?

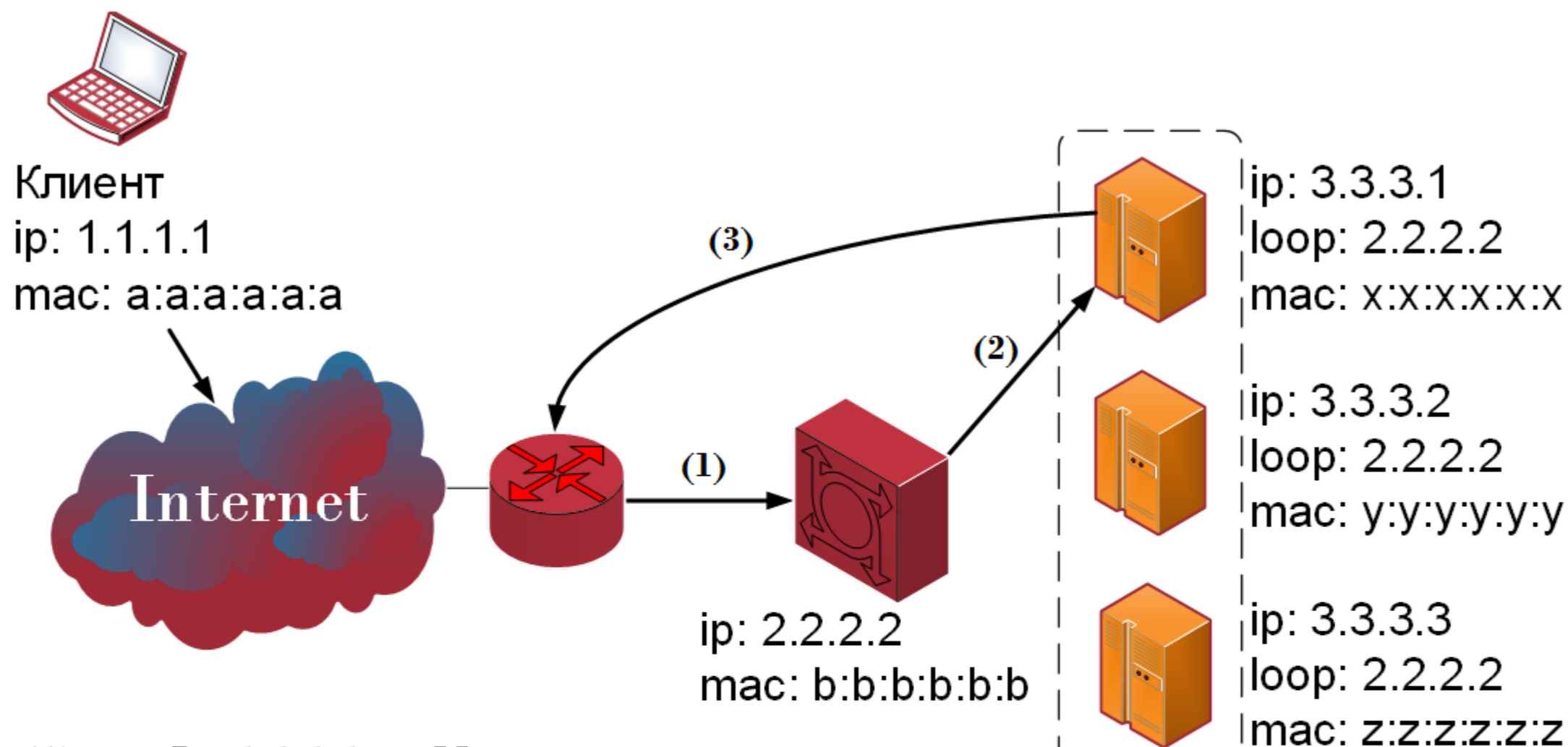


Зачем нужна балансировка?

- Основные цели:
 - Распределение нагрузки между серверами
 - Повышение отказоустойчивости
 - Защита от некоторых видов атак
- Предъявляемые требования к системе балансировки:
 - Справедливость
 - Эффективность
 - Сокращение времени выполнения запроса
 - Сокращение времени отклика
 - Предсказуемость
 - Равномерная загрузка ресурсов системы
 - Масштабируемость

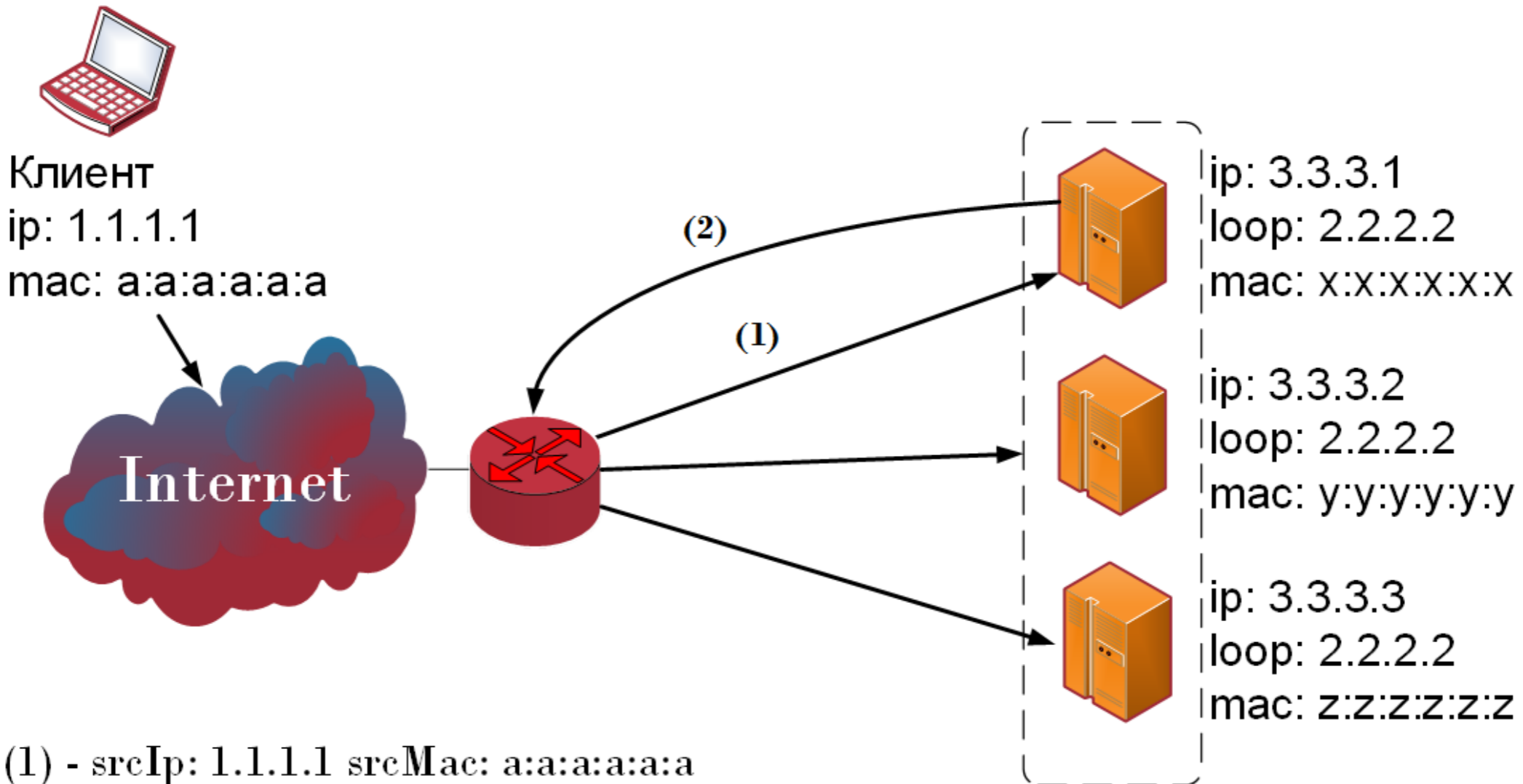


Балансировка на канальном уровне



- (1) - srcIp: 1.1.1.1 srcMac: a:a:a:a:a:a
dstIp: 2.2.2.2 dstMac: b:b:b:b:b:b
- (2) - srcIp: 1.1.1.1 srcMac: a:a:a:a:a:a
dstIp: 2.2.2.2 dstMac: x:x:x:x:x:x
- (3) - srcIp: 2.2.2.2 srcMac: x:x:x:x:x:x
dstIp: 1.1.1.1 dstMac: a:a:a:a:a:a

Балансировка на канальном уровне (shared address)



(1) - srcIp: 1.1.1.1 srcMac: a:a:a:a:a:a
dstIp: 2.2.2.2 dstMac: a:0:0:0:0:0 (отсутствует)

(2) - srcIp: 2.2.2.2 srcMac: x:x:x:x:x:x
dstIp: 1.1.1.1 dstMac: a:a:a:a:a:a

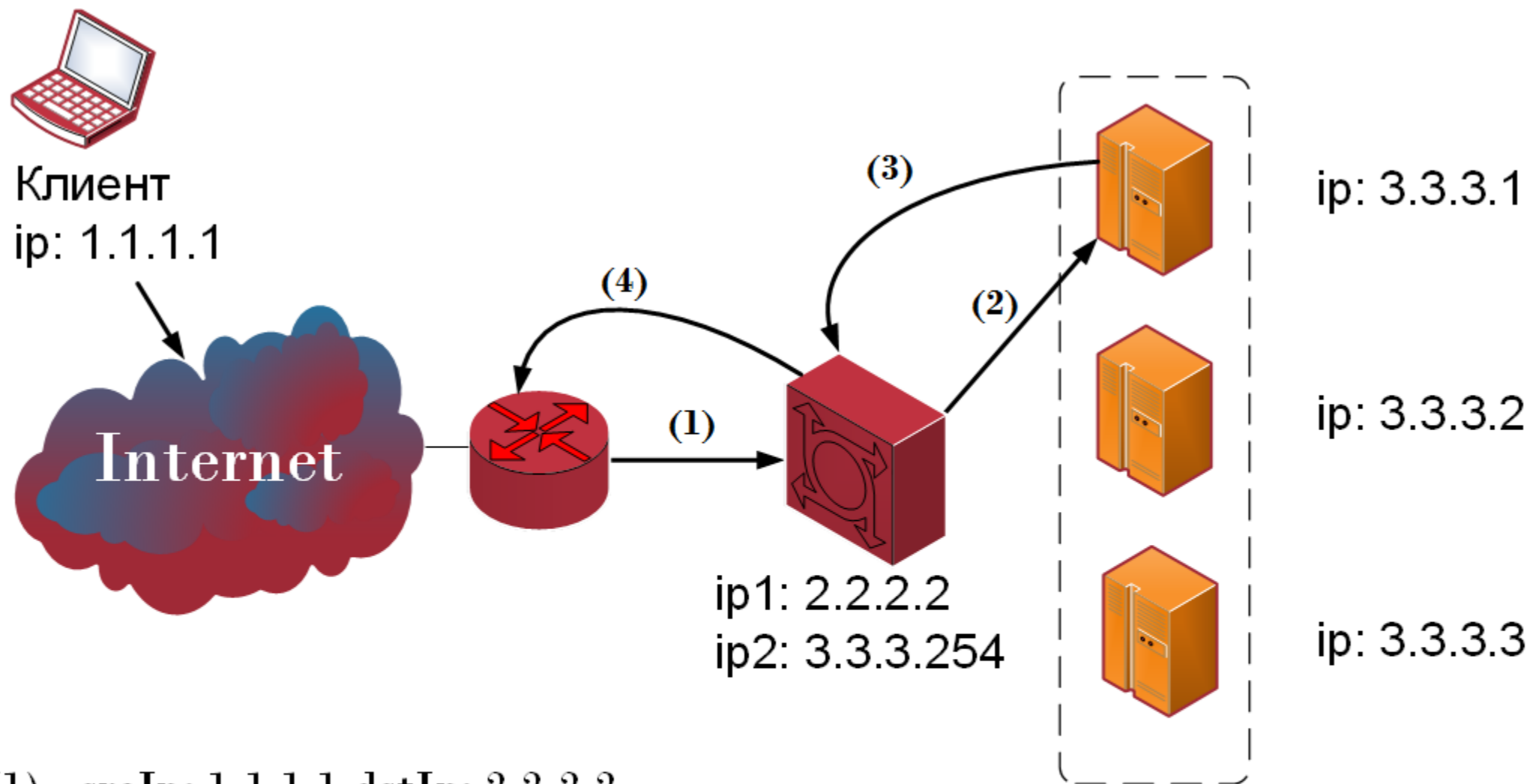
Серверы сайта
<http://mysite.ru>

Балансировка на канальном уровне

- Плюсы:
 - Не зависит от протоколов вышележащих уровней
 - Сокращение расходов за счет отказа от выделенного балансировщика
 - Обратный трафик в сторону клиента не нагружает балансировщик
 - Используется только один публичный адрес
 - Быстрое добавление и отключение сервера в кластере
- Минусы:
 - Необходимо размещать сервера в одном сегменте
 - Ограничение по входящей полосе (в случае с shared address входящий трафик попадает на все сервера одновременно)
- Решения: *Linux Virtual Server*



Балансировка на сетевом уровне



- (1) - srcIp: 1.1.1.1 dstIp: 2.2.2.2
- (2) - srcIp: 1.1.1.1 dstIp: 3.3.3.1
- (3) - srcIp: 3.3.3.1 dstIp: 1.1.1.1
- (4) - srcIp: 2.2.2.2 dstIp: 1.1.1.1

Балансировка на сетевом уровне

- **Преимущества:**

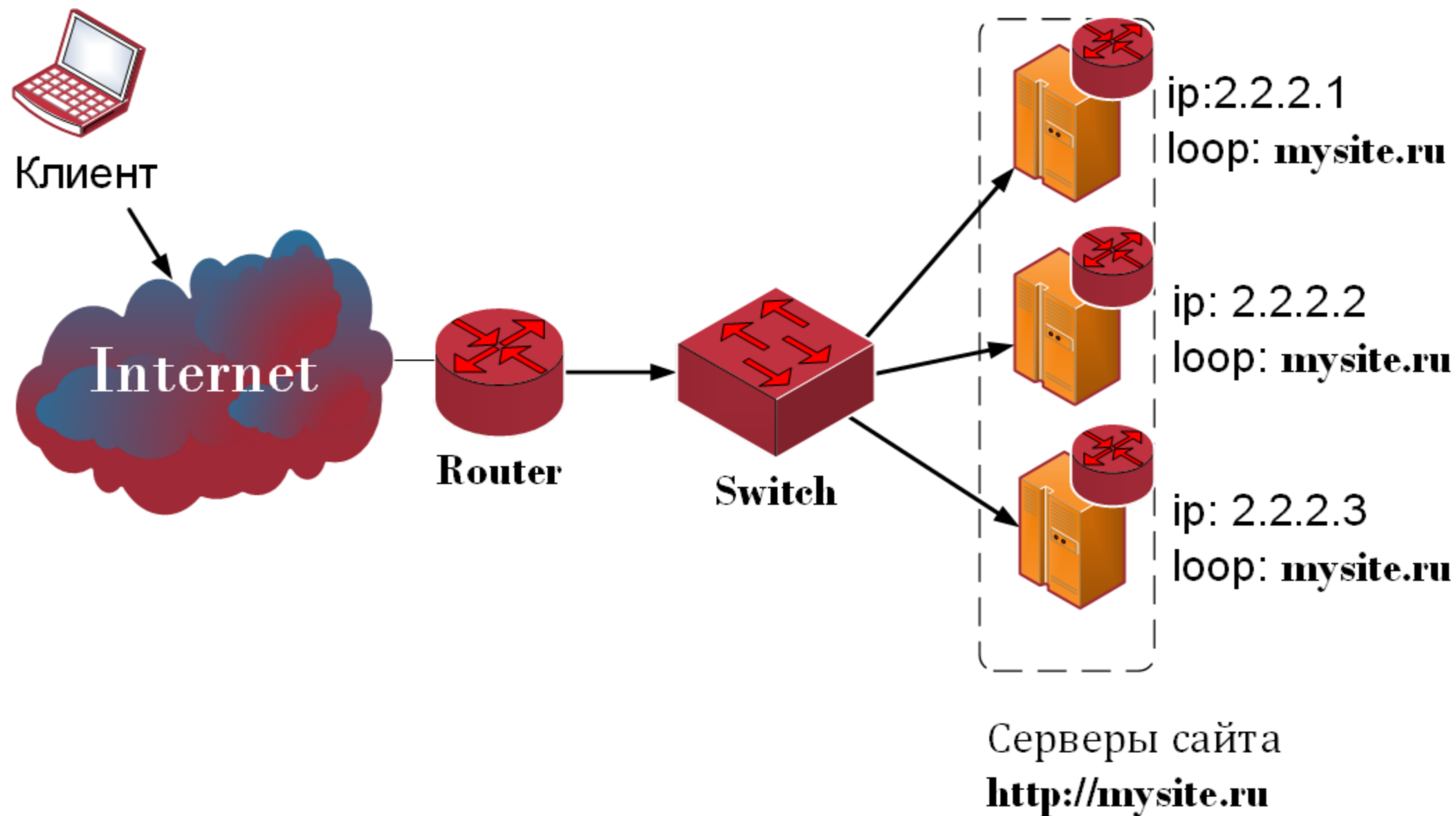
- Не зависит от протоколов высокого уровня
- Полная прозрачность работы для серверов
- Один публичный адрес

- **Недостатки:**

- Повышенная нагрузка на балансировщик за счет обратного трафика

- **Решения:** *Linux Virtual Server, различные аппаратные реализации*

Балансировка на транспортном уровне (ECMP)

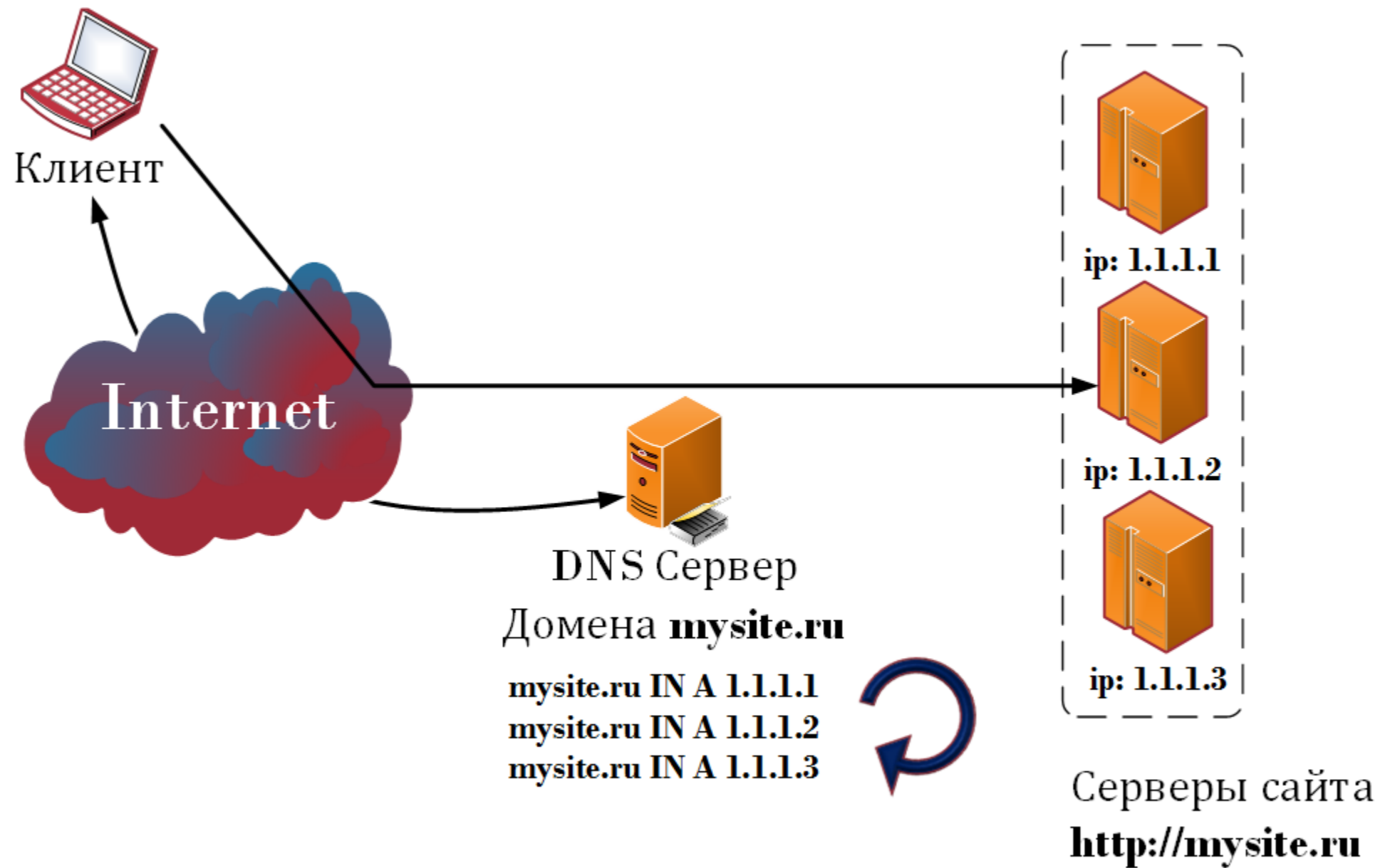


Балансировка на транспортном уровне

- Преимущества:
 - Не зависит от протоколов высокого уровня
 - Один публичный адрес
 - Отсутствие необходимости приобретать дополнительное оборудование
- Недостатки:
 - Необходимо ставить на сервера дополнительный софт
 - Отсутствует *server-affinity* (все соединения разрываются при добавлении/исключении сервера)
 - Ограничения количества ECMP на роутерах
 - Равномерность распределения нагрузки накладывает требование к выравниванию производительности серверов
 - Таймауты BGP приводят к распределению нагрузки на недоступный сервер
- Решения: *Cisco 3750X, 4500-X, 6500+Sup2T*



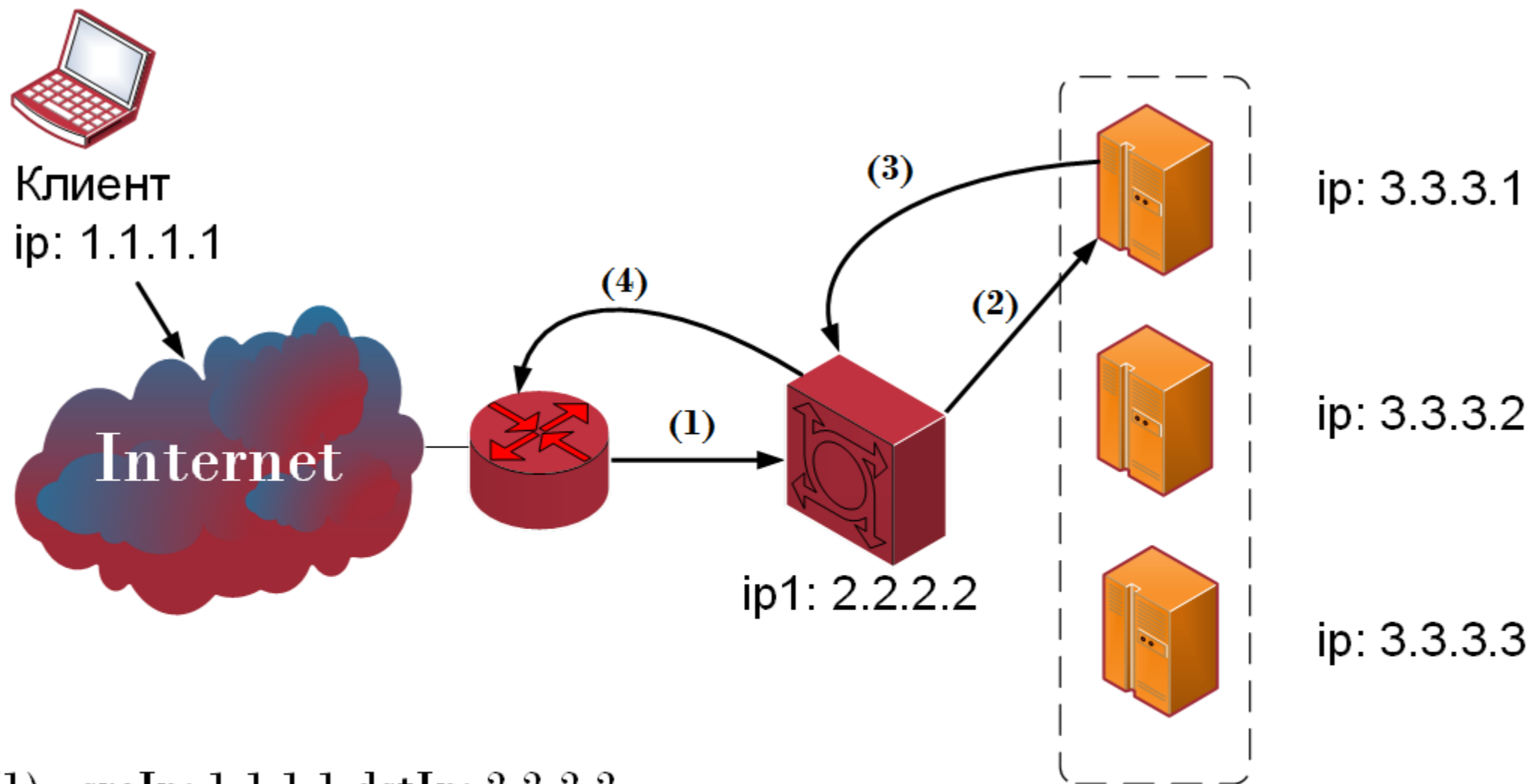
DNS Round Robin



DNS Round Robin

- **Сильные стороны:**
 - Абсолютная независимость от протокола высокого уровня
 - Независимость от нагрузки сервера благодаря кэширующим DNS-серверам
 - Универсальность (как локальная, так и глобальная балансировки)
 - Очень низкая стоимость решения и быстрый старт
- **Слабые стороны:**
 - Сложно отключать сервера (необходимо доп. резервирование по типу CARP)
 - Сложно распределять нагрузку в нужной пропорции
 - Ограниченность IP-адресов (т.к. каждый сервер должен иметь свой глобальный IP)
 - Необходимость держать двойной запас серверной мощности
- **Решения:** любой DNS сервер (например, Named)

Full proxy



- (1) - srcIp: 1.1.1.1 dstIp: 2.2.2.2
- (2) - srcIp: 2.2.2.2 dstIp: 3.3.3.1
- (3) - srcIp: 3.3.3.1 dstIp: 2.2.2.2
- (4) - srcIp: 2.2.2.2 dstIp: 1.1.1.1

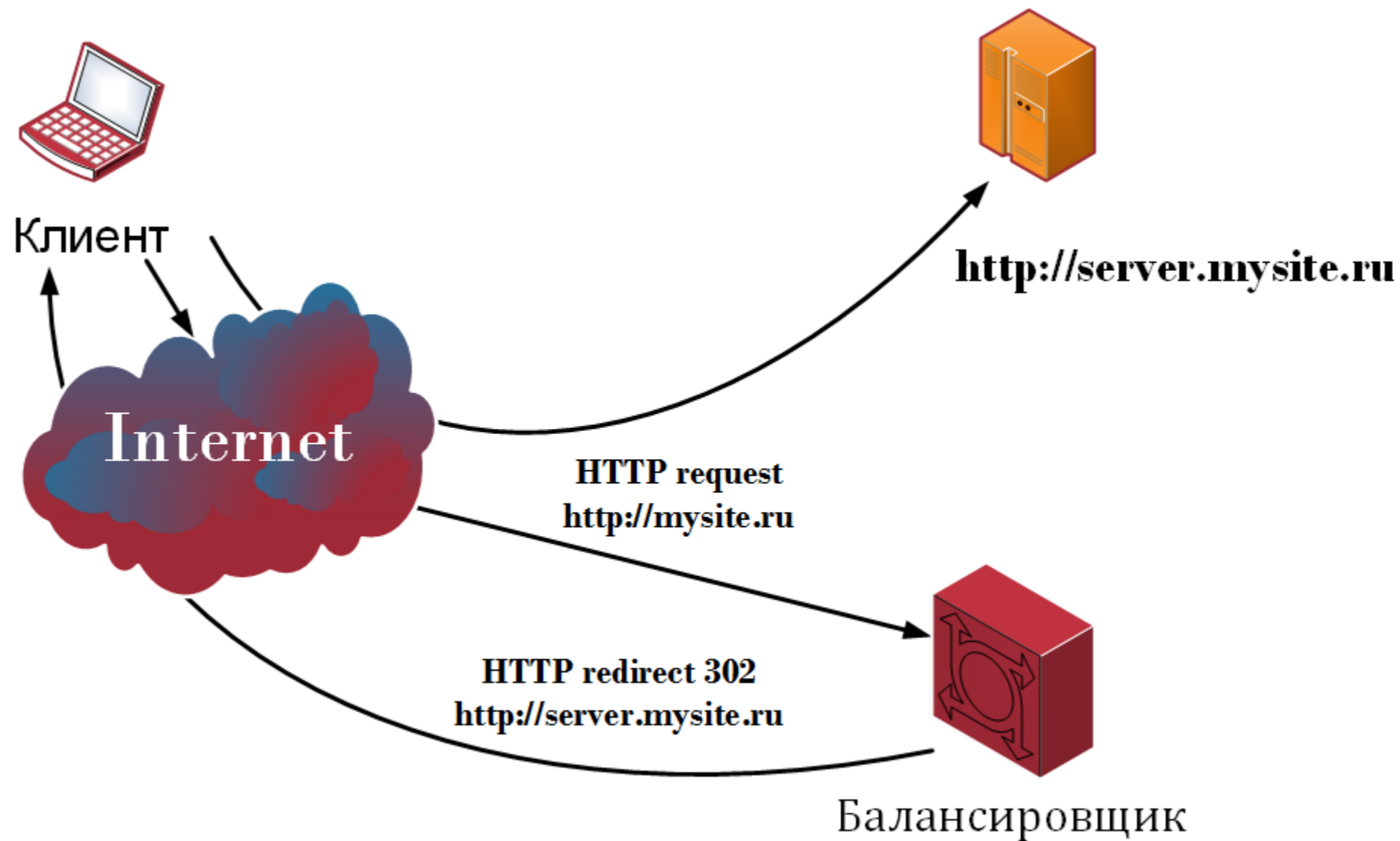
Серверы сайта
http://mysite.ru

Full proxy

- Плюсы:
 - Server-affinity (по определенной настройке cookie)
 - Распределение разных типов запросов к разным серверам
 - Возможность анализировать и модифицировать запросы
 - Фильтрация запросов по URL (защита от разных атак)
 - Самостоятельно определяют работоспособность каждого узла
- Минусы:
 - Необходимо балансировать нагрузку на сами балансировщики
 - Дополнительная точка отказа
 - Большое потребление ресурсов
 - Свой прокси для каждого протокола
- Решения: *HAProxy, nginx*



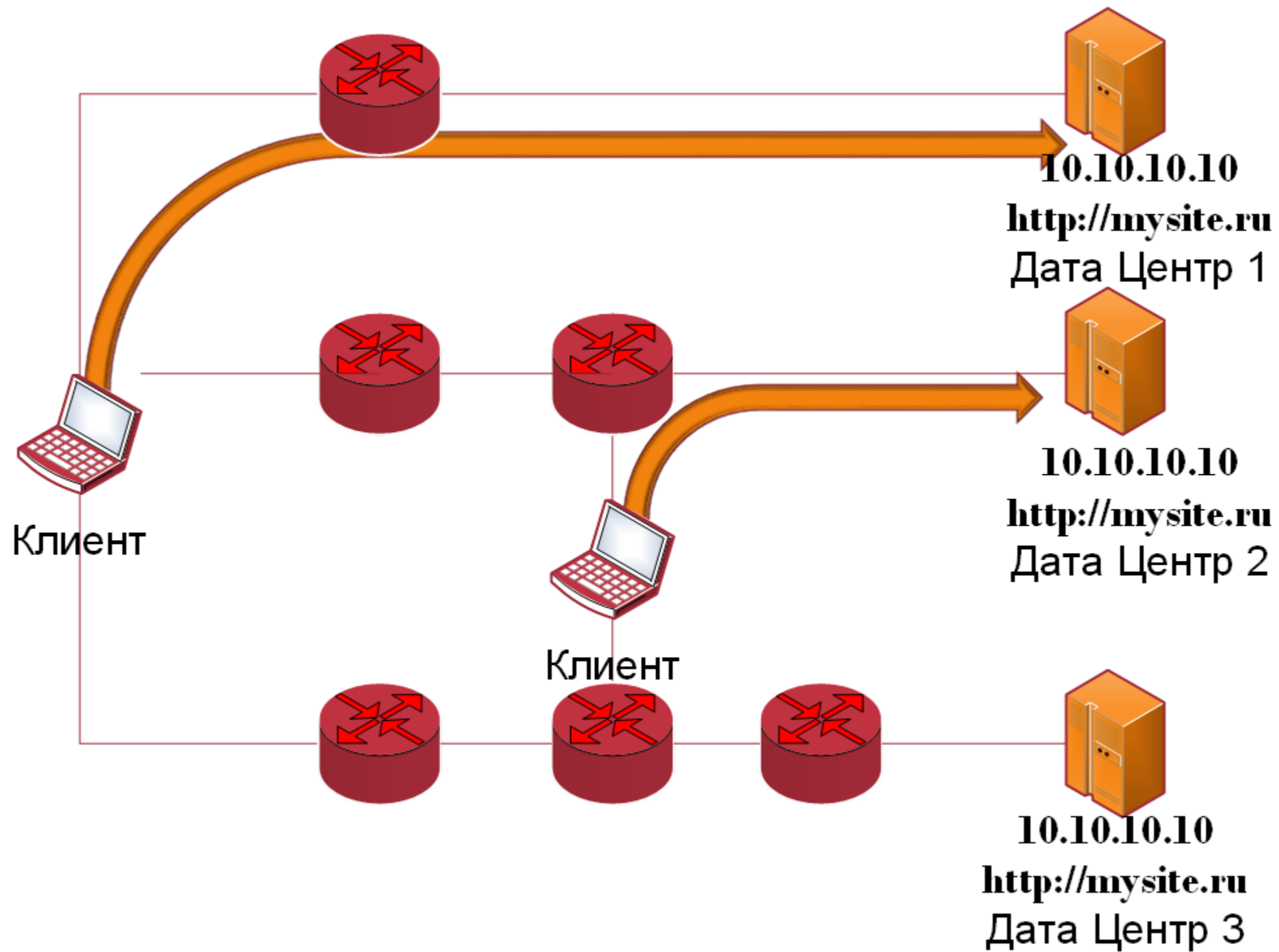
Redirect запросов (http)



Redirect запросов (http)

- Плюсы:
 - Распределение запросов по разным серверам за счет анализа запроса
- Минусы:
 - Достаточно малая применимость к протоколам высокого уровня
 - Увеличение времени отклика за счет обращения к редиректору
 - Два запроса к сервису на каждый запрос клиента
- Решения: *nginx*

Балансировка на базе Anycast



Балансировка на базе Anycast

- **Плюсы:**

- Минимальные задержки при обработке запросов
- Доставка трафика через немагистральные каналы (удешевление)
- Распределением нагрузки занимается сама сеть
- **Высокая отказоустойчивость**
- Легко добавлять и выводить из работы сервера

- **Минусы:**

- Возможность перестроения маршрутов (критично для TCP-сессий)
- Отсутствует возможность проконтролировать, с какого узла обслуживается пользователь (этим рулит сеть)
- Дорогое оборудование
- **Интересы ISP**

Алгоритмы распределения нагрузки

- Weighted Round Robin
- Least Connections
- Destination Hash Scheduling и Source Hash Scheduling
- Sticky Sessions

Несколько слов об интеграторах

- Cisco ACE
- F5 BigIP
- Alteon NG
- Cisco CSS



Благодарю за
внимание!

Вопросы?

