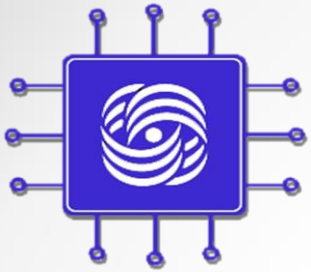


APPLIED
RESEARCH
CENTER FOR
COMPUTER
NETWORKS

SDN&NFV:
Технологии SDN/OpenFlow

Доп. главы Компьютерных сетей и телекоммуникации

к.ф.-м.н., м.н.с., Шалимов А.В.

ashalimov@lvk.cs.msu.su

@alex_shali

@arccnnews

Часть I: SDN

Уникальное время

“Россия и SDN – это идеальный брак, который должен состояться на небесах”

SDN уже здесь

Google перевел сеть между ЦОД на SDN в 2012 году, сейчас анонсирована внутренняя облачная платформа **Andromeda**.

Microsoft перевел сеть между ЦОД на SDN в конце 2013 года, на очереди публичное облако **Azure.**

NTT перевел всю свою сетевую инфраструктуру на SDN в 2013 году.

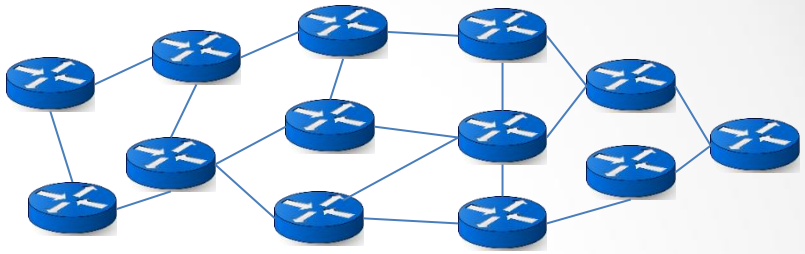
В июне 2015 года **AT&T** объявило SDN своим основным стратегическим направлением развития и переориентацию на разработку ПО.

Gartner: *“Рынок SDN решений к 2018 году достигнет объема $35 млрд”*.

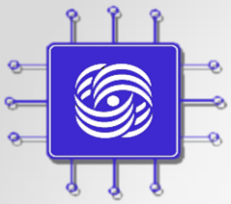
А что с SDN в России?

- «Ростелеком» начал работу над внедрением перспективных технологических направлений **«Программно-конфигурируемых сетей» (SDN)** и **«виртуализации сетевых функций» (NFV)**.
- «Ростелеком» и впредь намерен укреплять **технологическое лидерство**. Новые технологии позволят **упростить сетевую инфраструктуру** и **снизить стоимость эксплуатации сети**.
  - старший Вице-Президент по эксплуатации сетей связи «Ростелекома» Александр Цейтлин
- «Считаю, что технологии SDN и NFV позволят существенно сократить капитальные затраты и **ускорить ввод в строй новых сервисов**»
  - исполнительный директор по технической стратегии и архитектуре «Ростелекома» Эдуард Василенко
- «Ростелеком» разыскивает **стартапы**, которые занимаются **разработкой технологий** в области SDN и NFV
  - Руководитель направления Департамента управления венчурными активами компании Сергей Шлыков

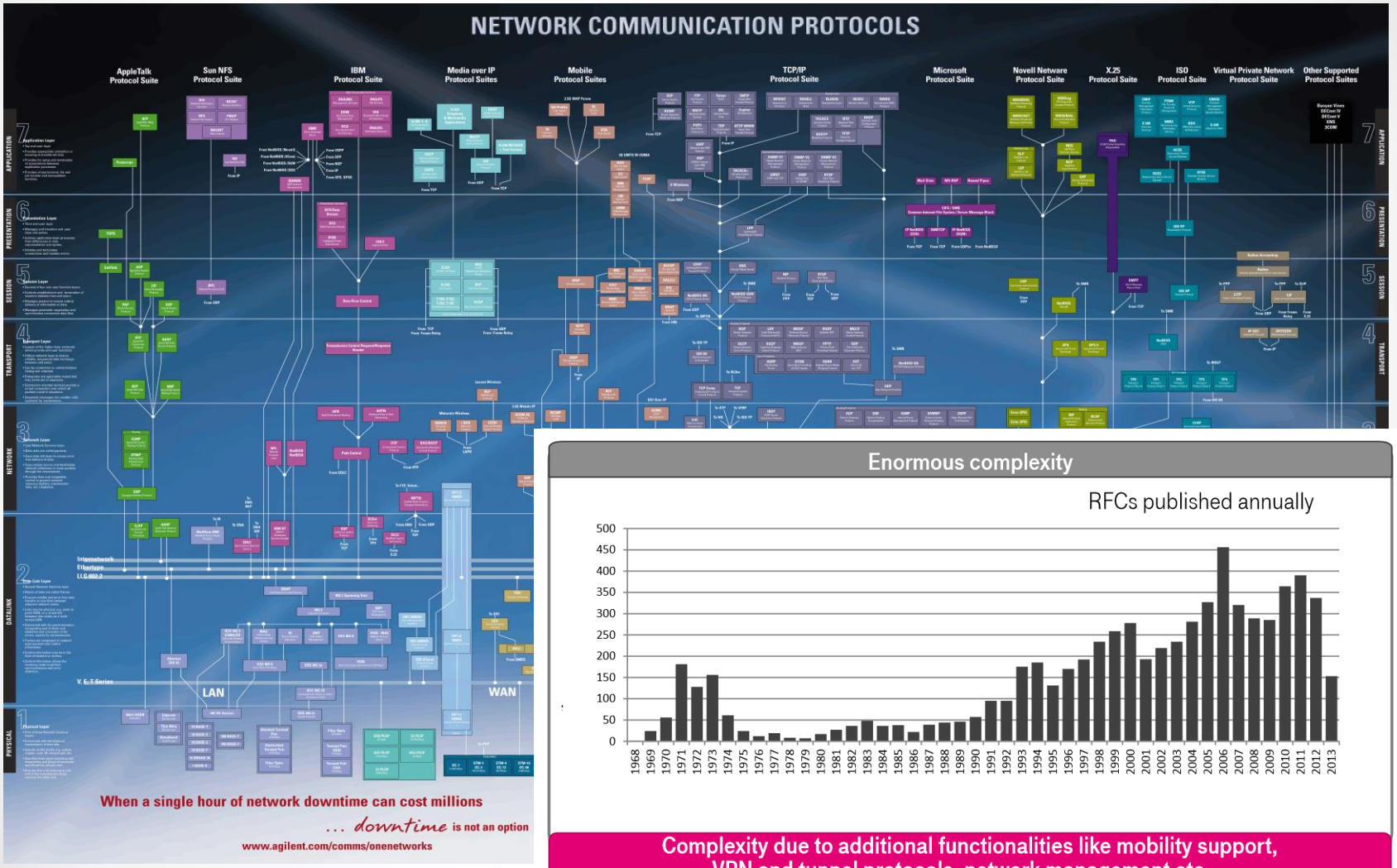
Проблемы традиционных сетей



- Зависимость от производителя
- Ошибки в реализациях сетевых протоколов
- Миллионы строк закрытого проприетарного кода (6000+ RFC)
- Высокая стоимость оборудования
- Высокая стоимость эксплуатации
- Сложность управления большими сетями
- Сложность отладки
- “Закрытость” оборудования и программного обеспечения
- Сложность внедрения новых идей
- Неэффективность использования аппаратных ресурсов, энергоэффективность



Постоянный рост сложности



Source: http://www.telegeography.com/products/ip_transit/index.php; http://www.ietf.org/

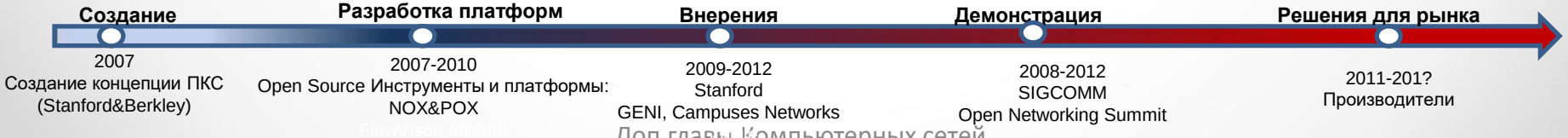
Основные принципы SDN

Программно-**К**онфигурируемые **С**ети (**S**oftware **D**efined **N**etworking/**SDN**) – это разделение плоскости передачи и управления данными, позволяющее осуществлять программное управление плоскостью передачи, которое может быть физически или логически отделено от аппаратных коммутаторов и маршрутизаторов

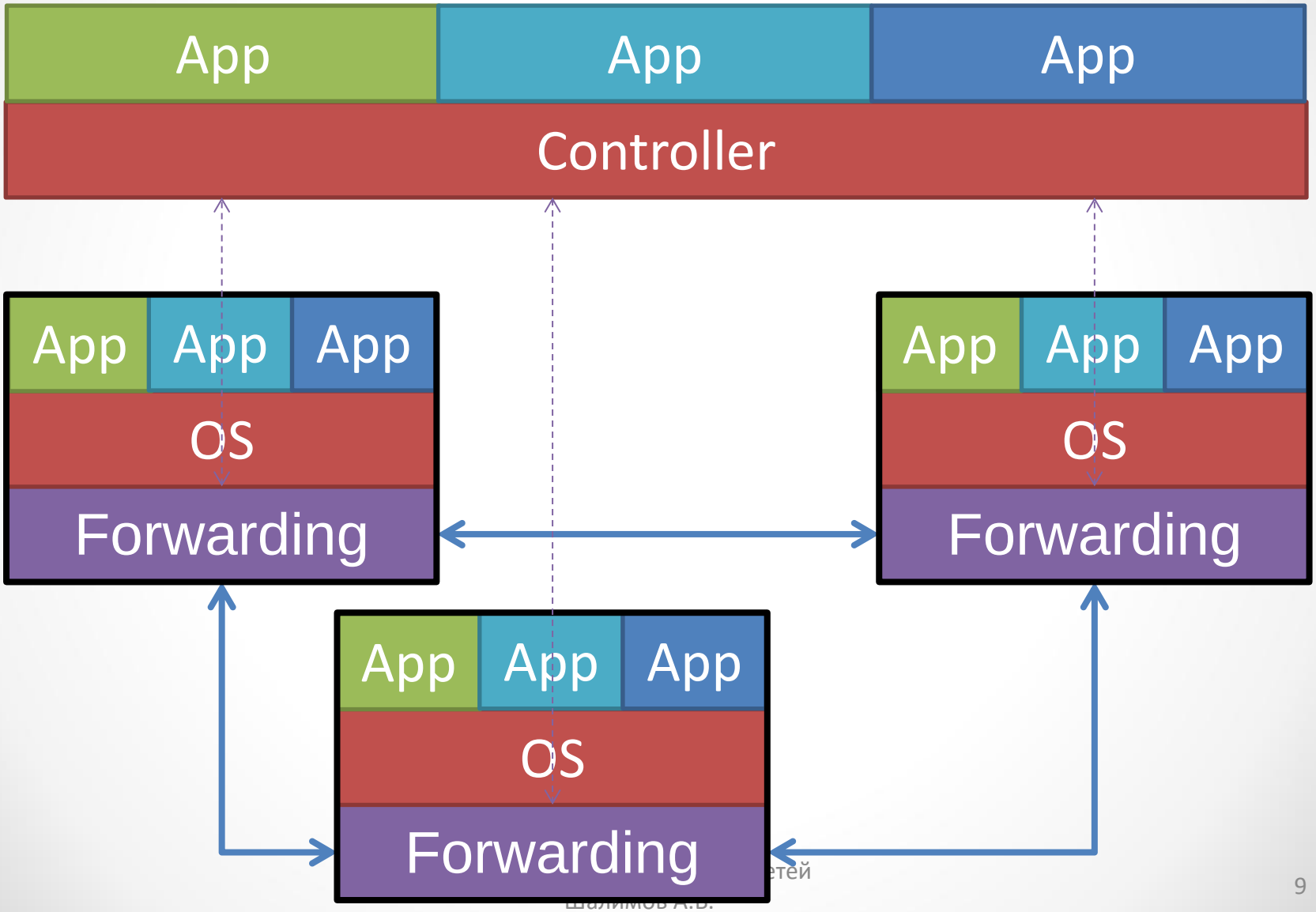
1. Отделить управление сетевым оборудованием от управления передачей данных за счет создания специального программного обеспечения.
2. Перейти от управления отдельными экземплярами сетевого оборудования к управлению сетью в целом.
3. Создать интеллектуальный, **программно-управляемый интерфейс** между сетевыми приложениями и транспортной сетью.

На самом деле ПКС – как четвертое поколение сотовых телефонов, только в сфере сетевых технологий :

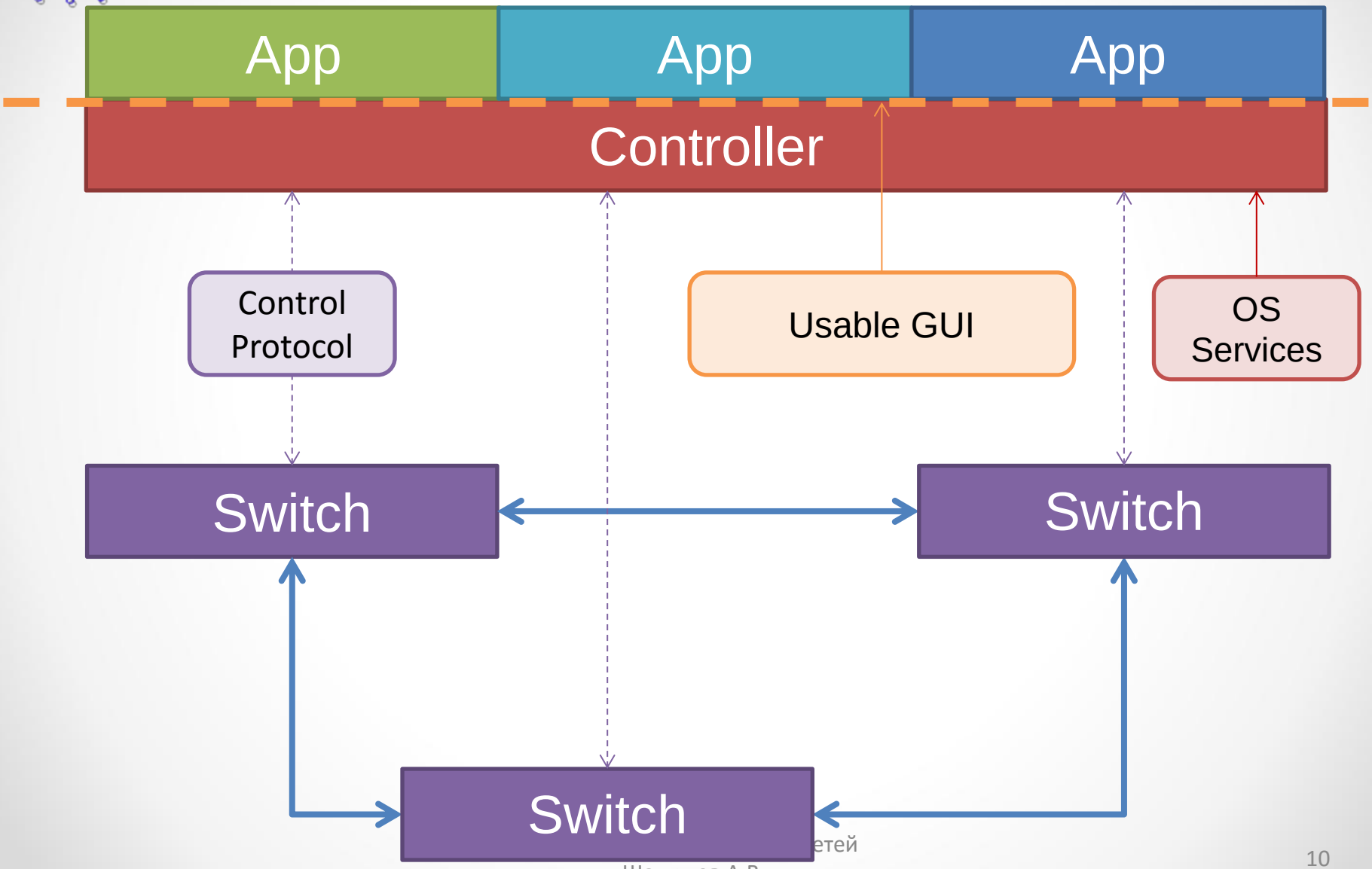
- Новые инструменты и функции;
- Простота администрирования;
- Открытость инновациям и экспериментам;
- Революция на ИТ-рынке



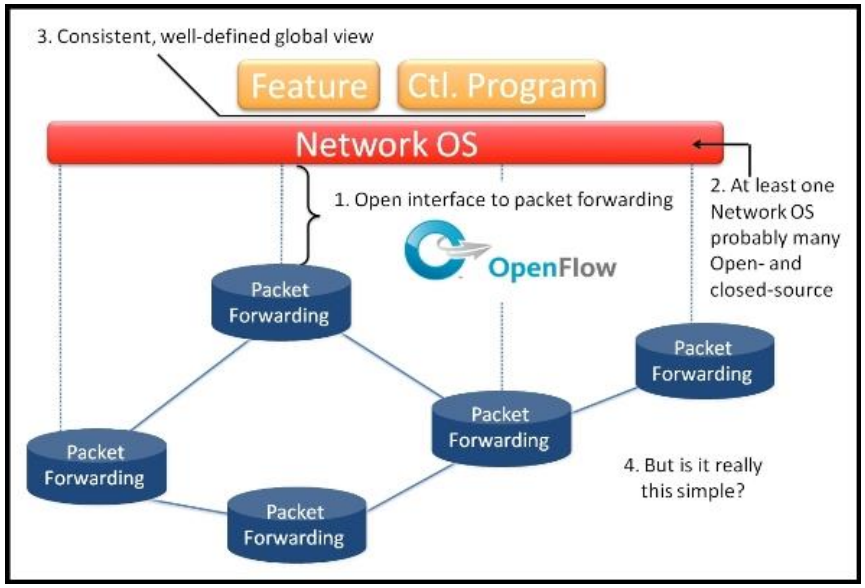
Переход к SDN



Архитектура SDN

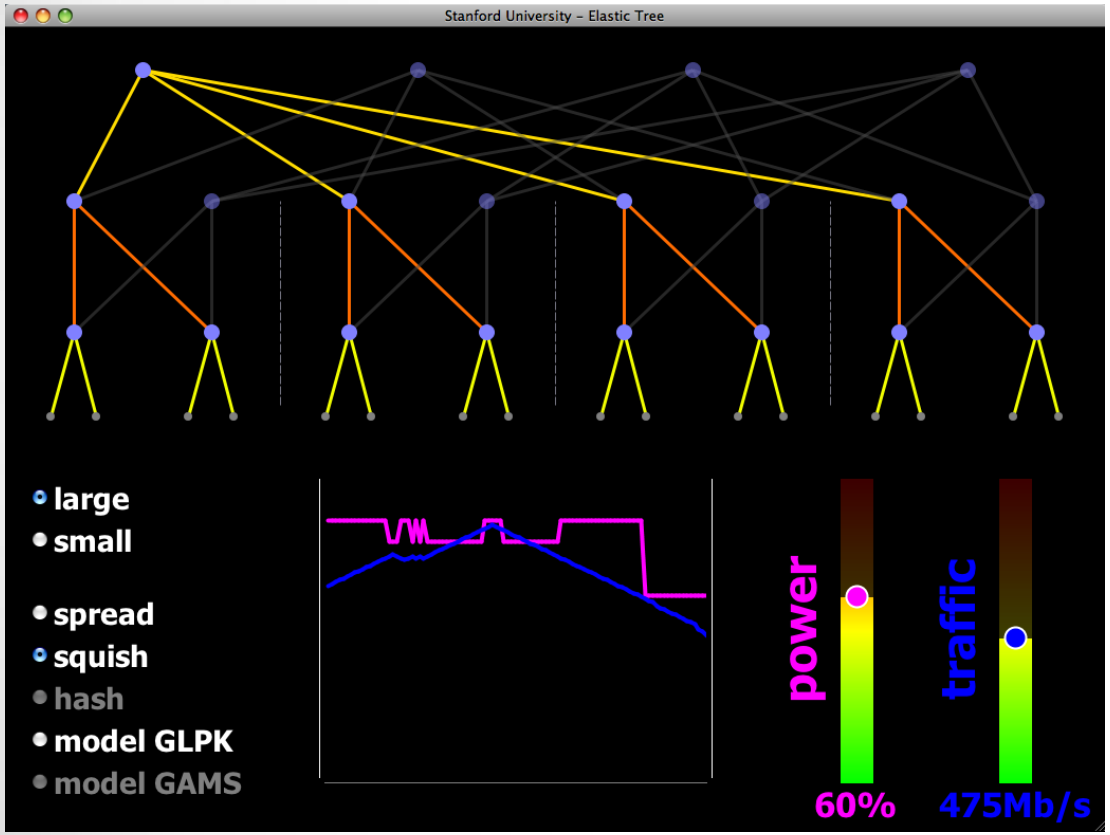


Достоинства



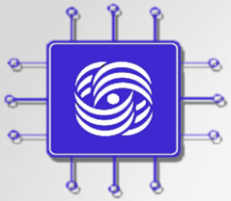
- **Удешевление оборудования (CAPEX)**
- **Облегчение управления сетью (OPEX)**
- **Программируемость, открытость, инновации**

Пример применения



Уменьшение энергопотребления в ЦОД

- Отключение неиспользуемых коммутаторов и каналов на основе собранной информации о сети
- ElasticTree (Stanford): сокращение энергопотребления до 60%
- Применение в Google

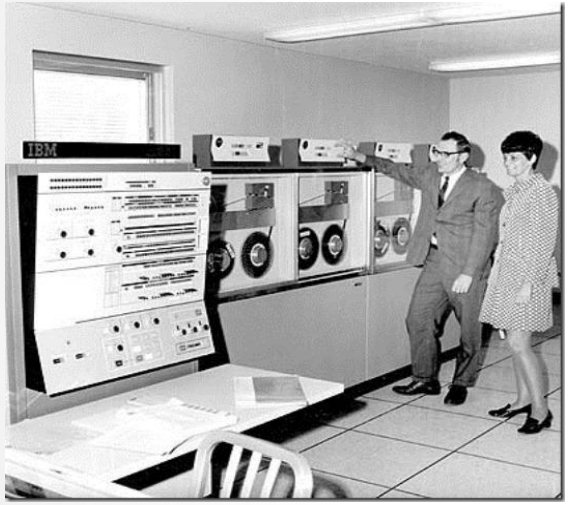


Абстракция

- Курсы по Операционным системам учат фундаментальным принципам:
  - Примитивы синхронизации, потоки, исключения, файловая система и т.д.
  - Новые языки программирования, операционные системы
- Курса по Сетевым технология учат куче протоколов
  - TCP, UDP, ARP, MPLS, GRE, BGP, OSPF, IS-IS, LDP, RSVP, PIM, ….
  - Отсутствие фундаментальных принципов, только руководства по эксплуатации сетей
  - Алгоритмы маршрутизации одни и теже много лет, управление сетью примитивно

Абстракции в IT

Медленно развивающаяся, закрытая, дорогая система. Малый рынок сбыта



Специализированные программы

Специализированная операционная система

Специализированная аппаратура

Быстрое внедрение инноваций
Открытые интерфейсы
Большой рынок сбыта

Приложения

Открытый интерфейс

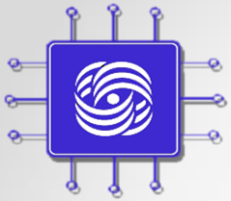
Операционные системы

Windows iOS

Открытый интерфейс

Микропроцессоры

intel AMD QUALCOMM



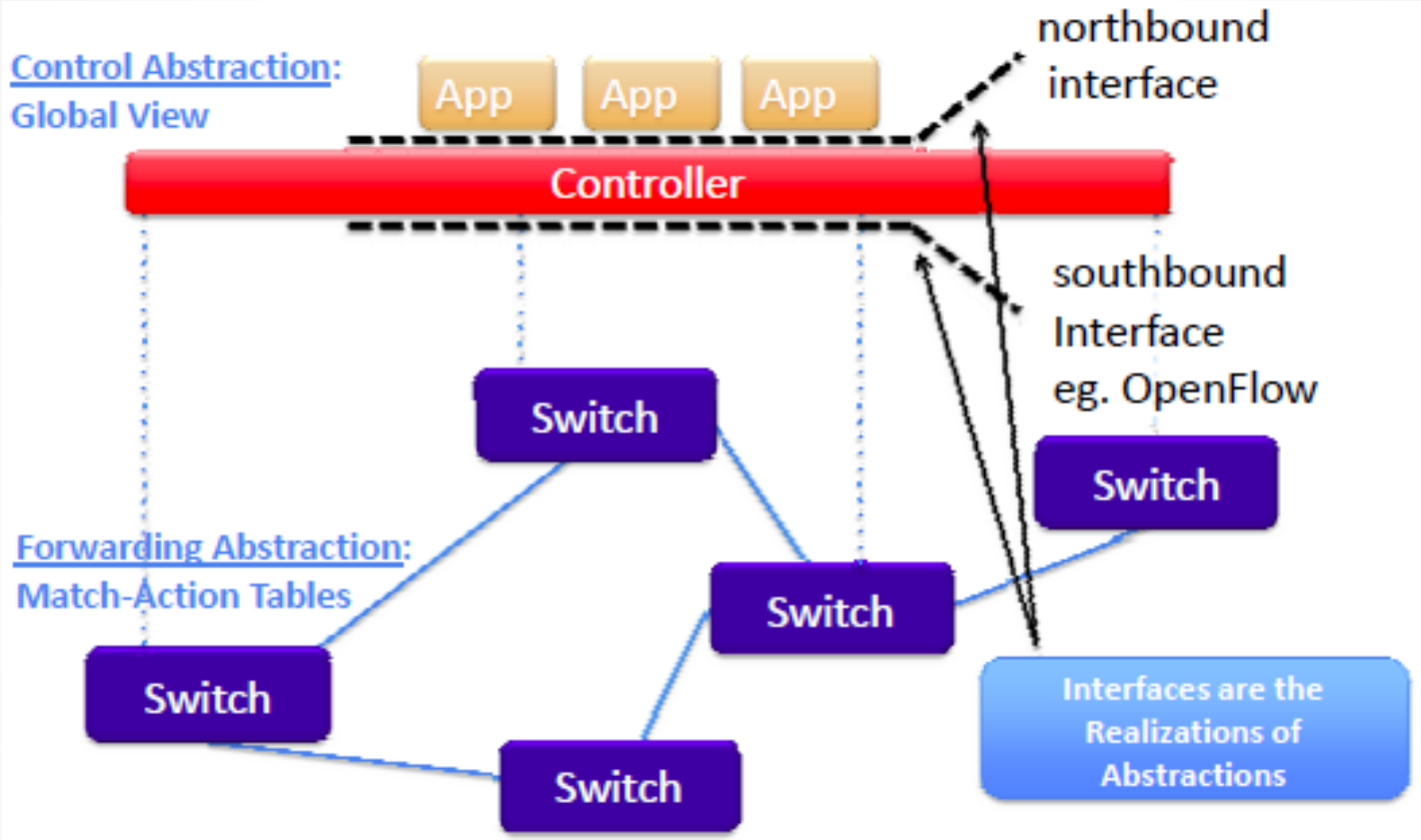
Абстракция в ЯП

- **Машинные языки программирования - нет никакой абстракции**
- **Высоко-уровневые языки программирования, ОС + другие абстракции**
  - Структуры данных, функции, переменные, файлы, виртуальная память, ...
- **Современные языки программирования – еще больше абстракций**
  - Объекты, нити, семафоры, сборщик мусора

Абстракции упрощают программирование: проще писать, проще поддерживать, думать об алгоритме

Можно ли такое же получить для компьютерных сетей?

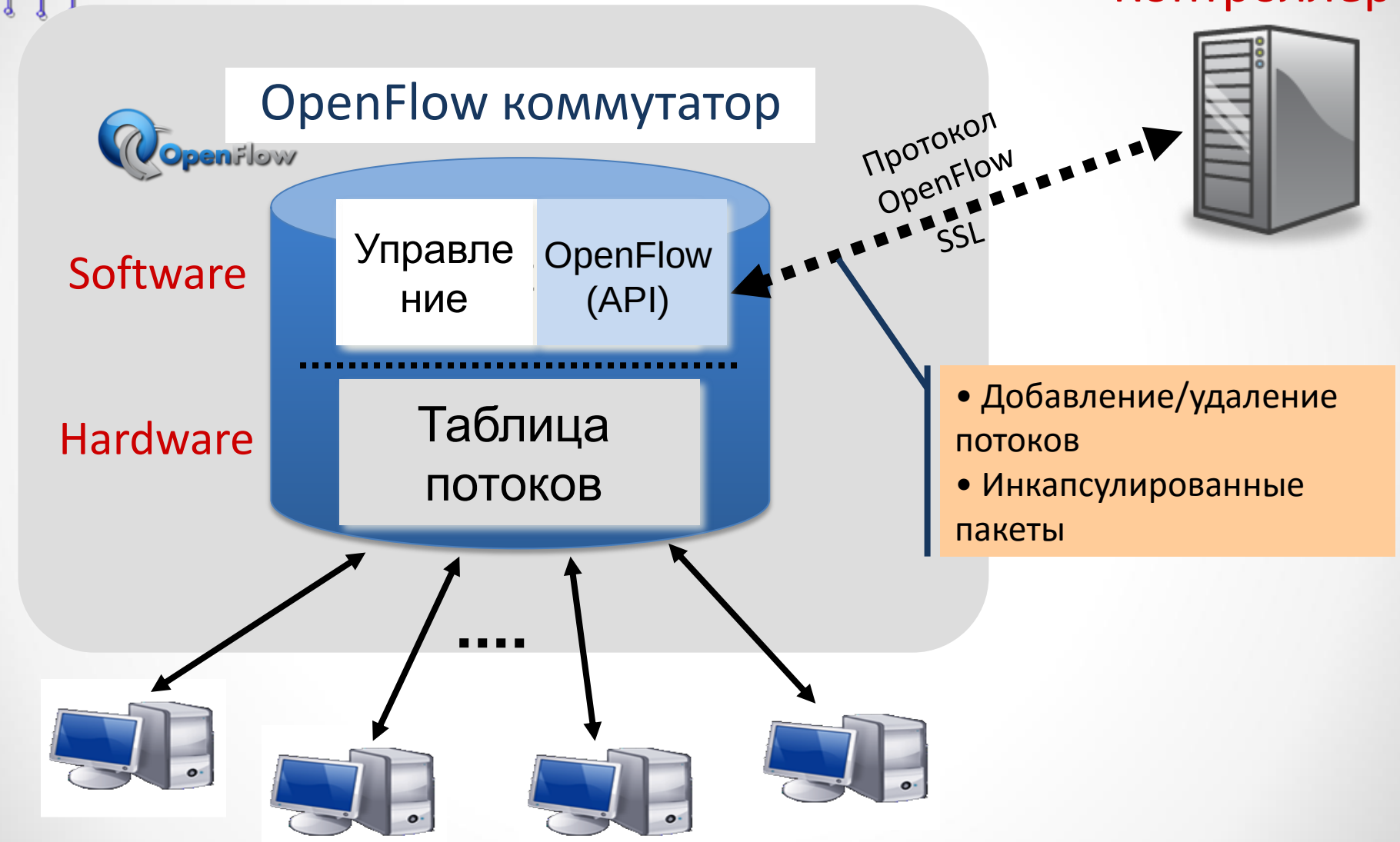
Абстракция в SDN?



- Абстракция уровня управления
- Абстракция уровня передачи данных

Часть II: OpenFlow

OpenFlow


Контроллер
OpenFlow коммутатор
OpenFlow
Протокол
OpenFlow
SSL
Software
Hardware
Управление
OpenFlow (API)
Таблица потоков
• Добавление/удаление потоков
• Инкапсулированные пакеты
....

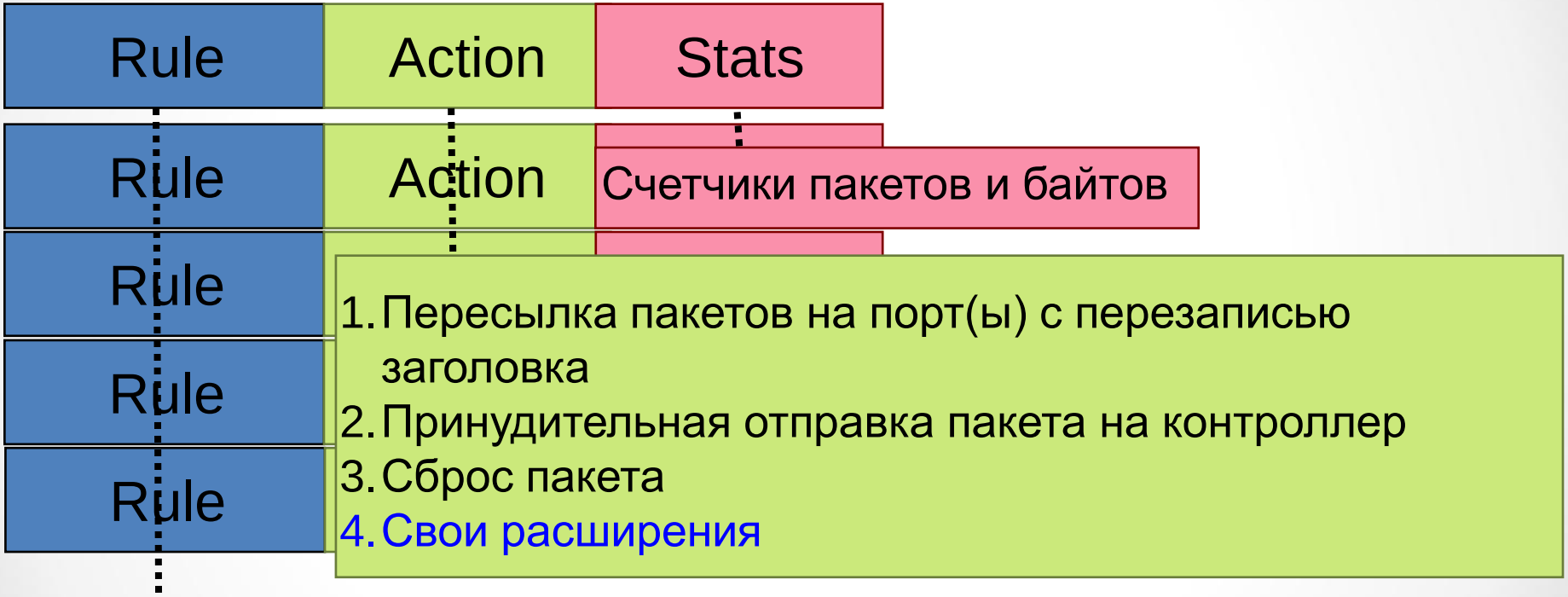
OpenFlow протокол

Поддерживается три типа сообщений:

- Сообщения контроллер-коммутатор
  - Конфигурирование коммутатора
  - Управление и контроль состояния
  - Управление таблицами потоков
  - Features, Configuration, Modify-State (**flow-mod**), Read-State (multipart request), **Packet-out**, Barrier, Role-Request
- Симметричные сообщения
  - Отправка в обоих направлениях
  - Обнаружение проблем соединения контроллера с коммутатором
  - Hello, Echo
- Ассиметричные сообщения
  - Отправка от коммутатора к контроллеру
  - Объявляют об изменении состояния сети, состояния коммутаторов
  - **Packet-in**, flow-removed, port-status, error

OpenFlow 1.0

Flow Table



+ маска по полям

Примеры правил OpenFlow

Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6

Flow Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
port3	00:20..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6

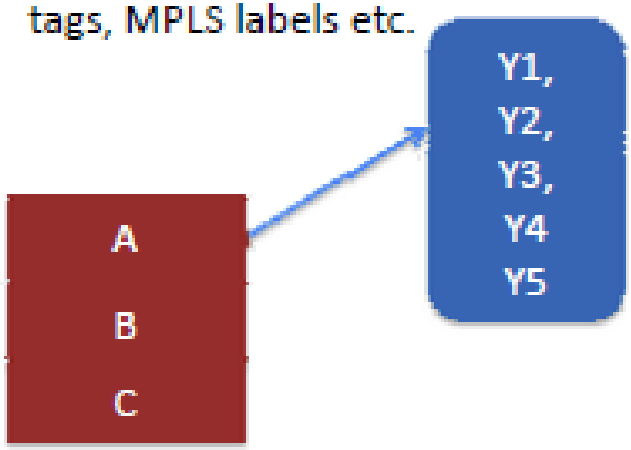
Firewall

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	*	*	*	22	drop

Чем плохо одна таблица?

- Table space explosion

A, B, C, Y could be MAC or IP addresses, VLAN tags, MPLS labels etc.

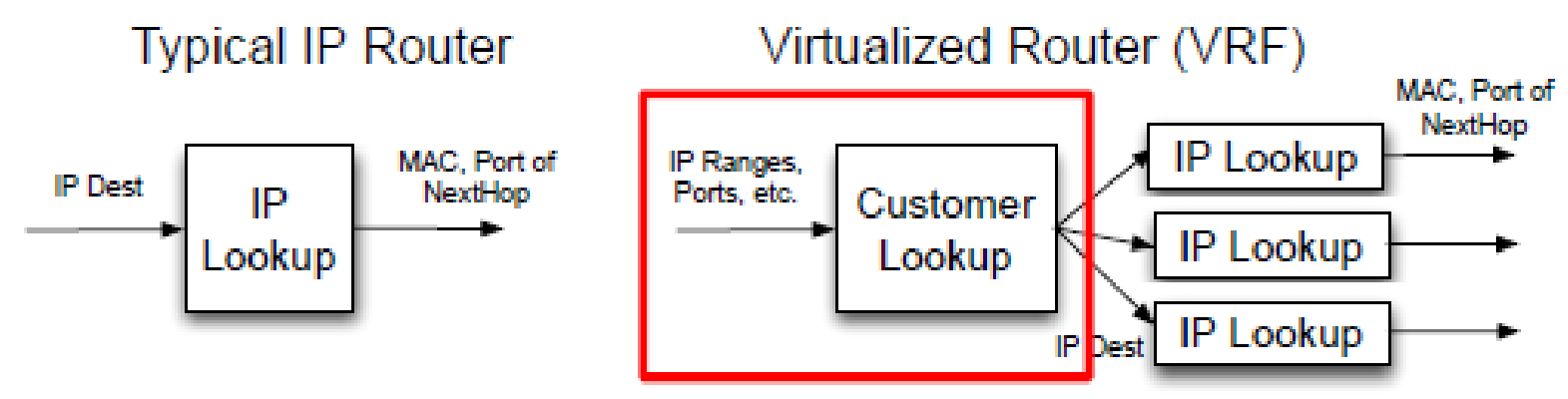


OF 1.0 Single Table

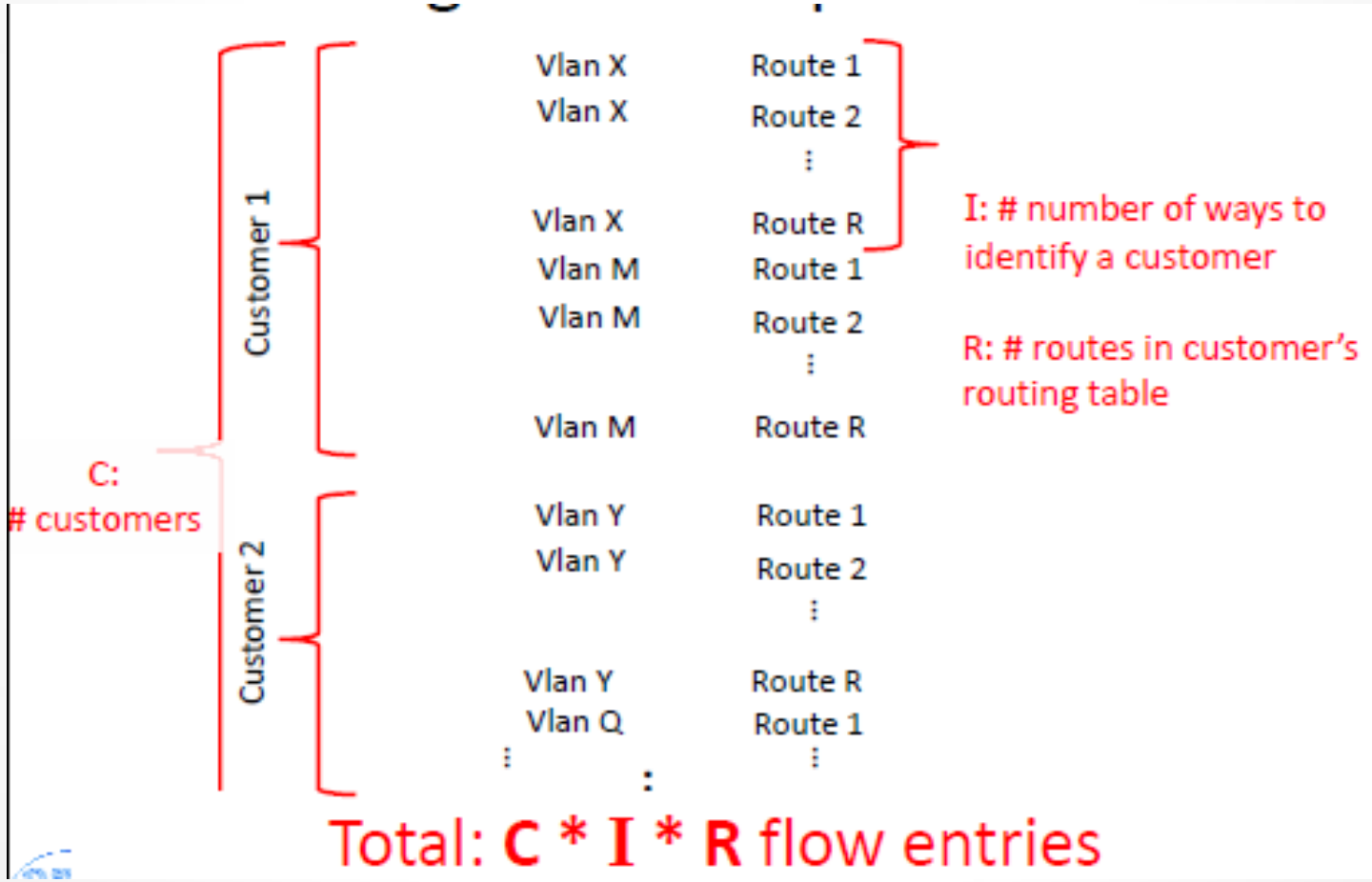
A, Y1
A, Y2
A, Y3
A, Y4
A, Y5
B, Y1
B, Y2
B, Y3
B, Y4
B, Y5
C, Y1
C, Y2
C, Y3
C, Y4
C, Y5

Single-table abstraction may use table space inefficiently compared to multiple tables

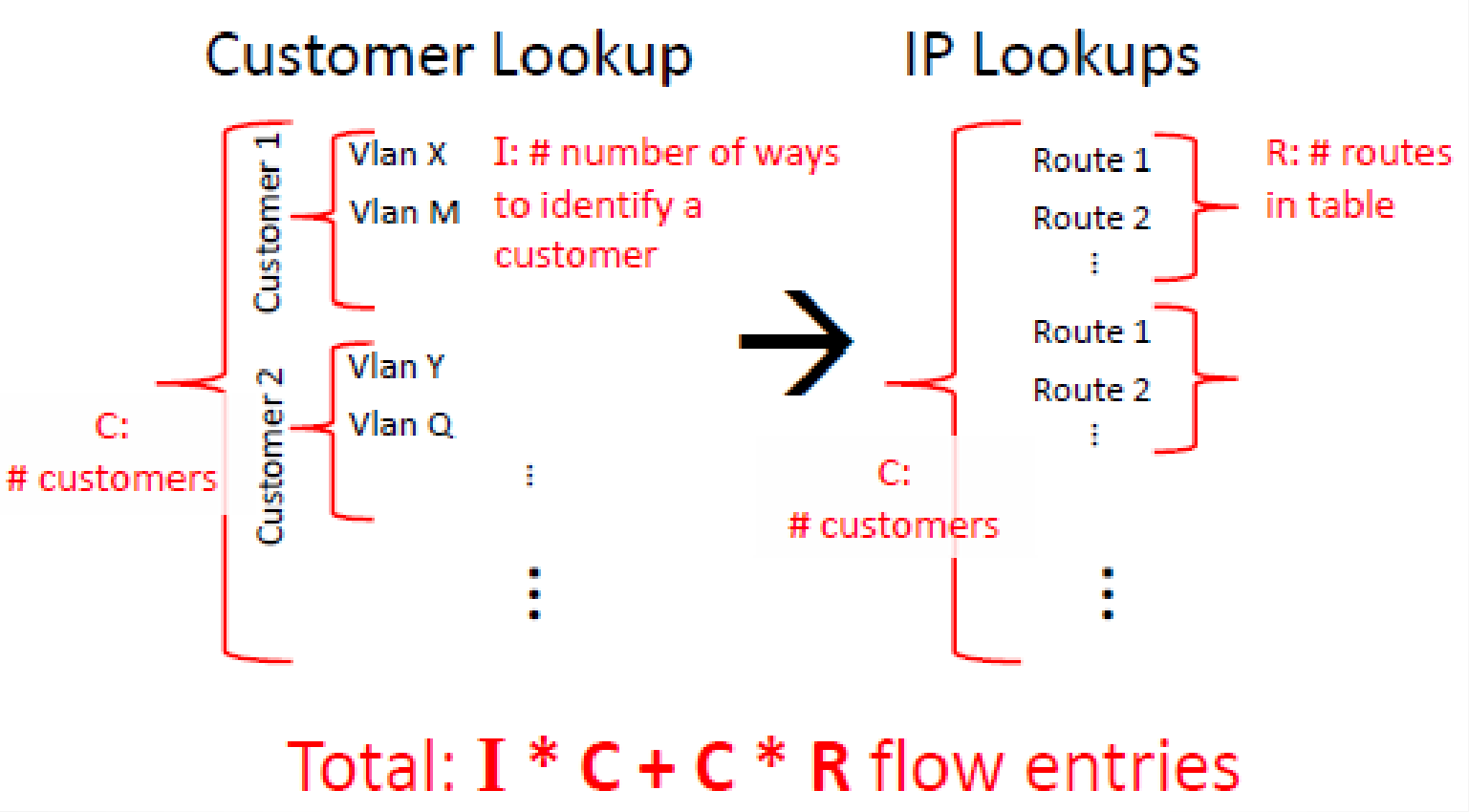
Пример: Virtual Routing and Forwarding (VRF)



VRF на одной таблице

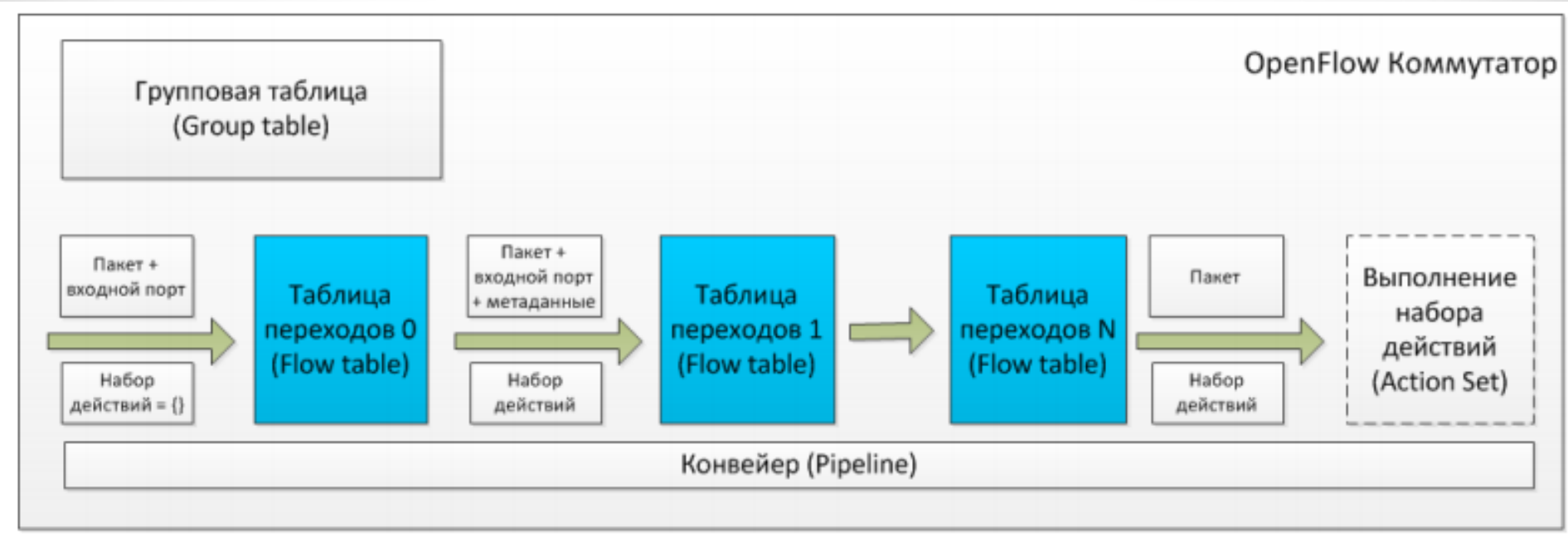


VRF на двух таблицах



- I = 10 vlan/customer; C = 100 customer; R = 1K IP addrs
- 1M vs 101K записей в таблицах

OpenFlow 1.1



- Продвижение пакета только вперед
- Переход: модификация пакета, обновление набора действий, обновление метаданных

Групповые таблицы

Идентификатор группы	Тип группы	Счётчики	Контейнеры действий

Определены следующие типы групп:

All - выполняются все контейнеры действий в группе.

Select - выполняется только один контейнер действий в группе.

Indirect - выполняется один определённый контейнер действий в группе.

Fast failover - выполняется первый существующий (живой) контейнер действий.

- Экономия места для одинаковых действий
- Также для реализации сетевых механизмов:
  - Multicast
  - ECMP
  - Active/Standby маршруты

Meter table

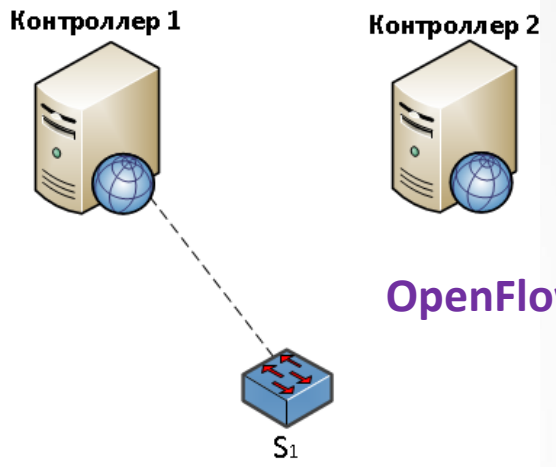
- Для реализации QoS и ограничения скорости
  - Для каждого потока или группы потоков
  - Следит за превышение значений счетчиков
  - Действия: **drop** или **dscp remark**

Meter Identifier	Meter Bands	Counters

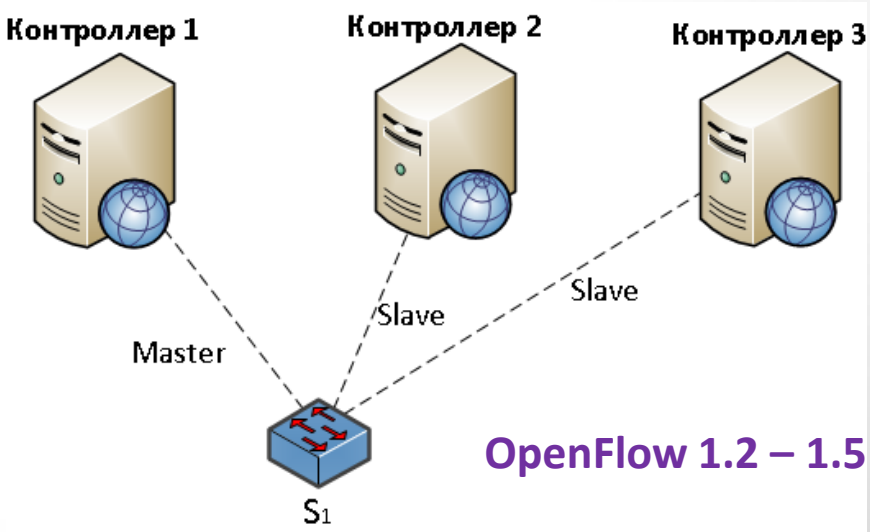
Band Type	Rate	Counters	Type specific arguments

Несколько контроллеров

- Протокол OpenFlow 1.2:
  - Множество контроллеров
  - Механизм ролей
  - **Роли:** Master, Slave, Equal
  - **По умолчанию:** контроллер находится в роли Equal для коммутаторов.
  - **Смена роли:** OFPT_ROLE_REQUEST
  - **Распределение ролей:** возложено на контроллеры.

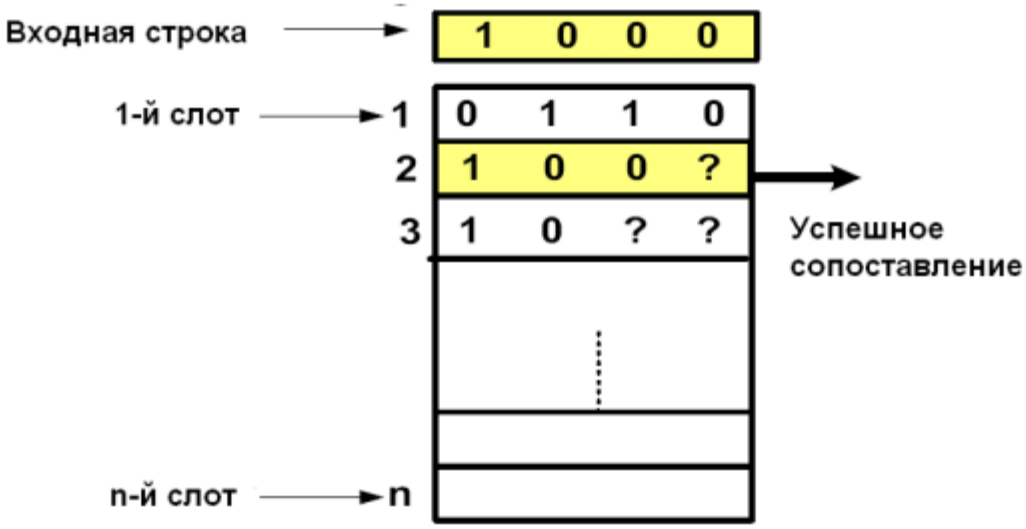


OpenFlow 1.0, 1.1

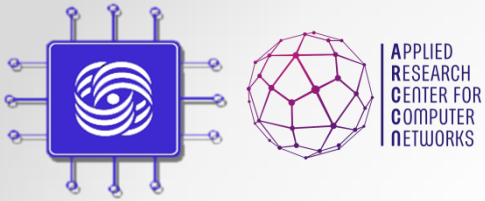


OpenFlow 1.2 – 1.5

TCAM троичная ассоциативная память



- Множество нумерованных слотов
- Три возможных значения каждого бита: “0”, “1” и “?”
- Ширина TCAM (длина слота) – настраиваемый параметр
- На вход подается битовая строка
- TCAM выдает номер первого слота с успешным сопоставлением
- Фиксированное время каждого такта работы TCAM



OpenFlow контроллер

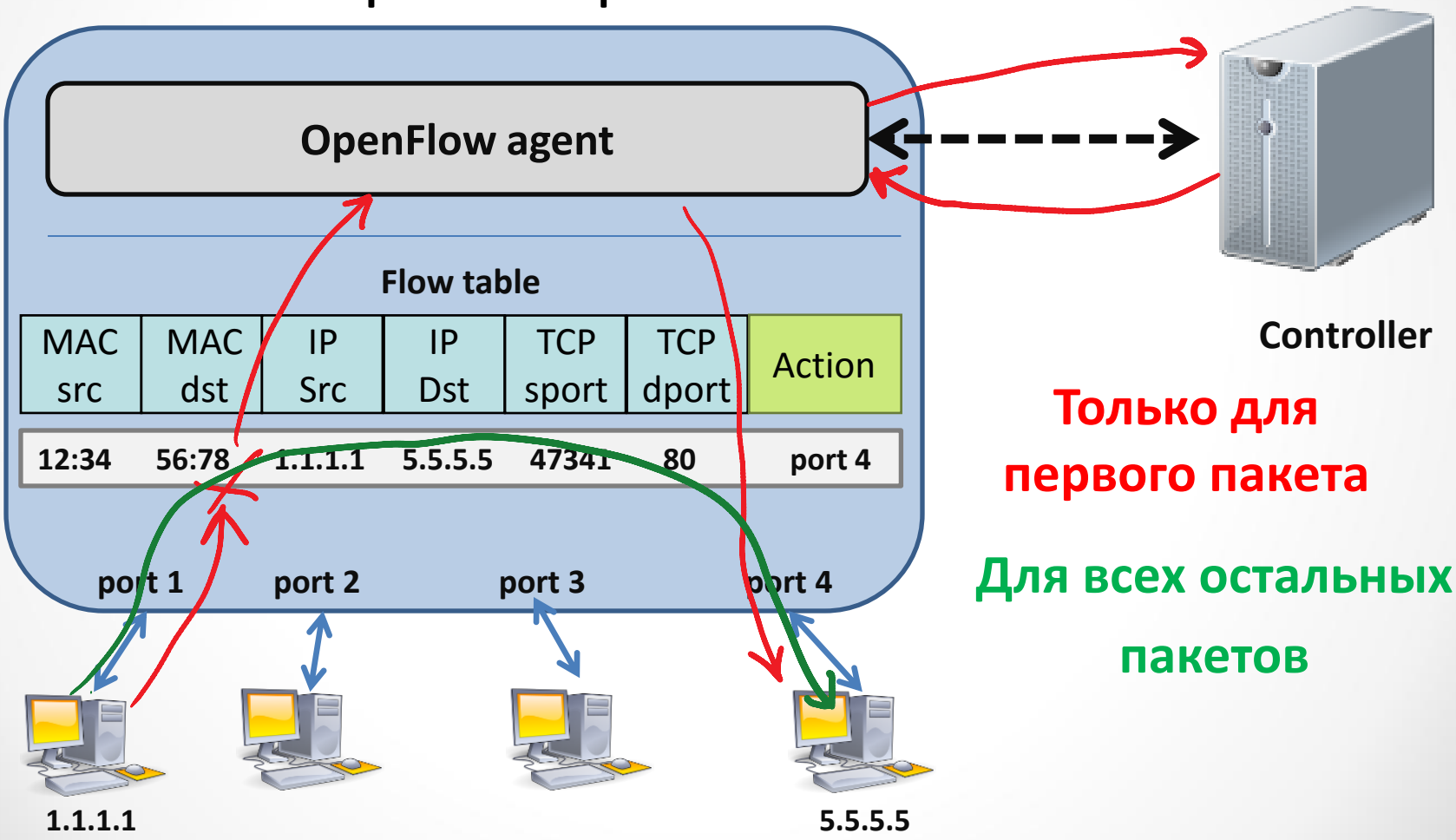
- Программа, TCP/IP сервер, ожидающий подключения коммутаторов
- Отвечает за обеспечение взаимодействие приложения-коммутатор.
- Предоставляет важные сервисы (например, построение топологии, мониторинг хостов)
- API сетевой ОС или контроллер предоставляет возможность создавать приложения на основе **централизованной модели программирования**.

Список OpenFlow контроллеров

- Их действительно много
  - Nox, Pox, MUL, Ruy, Beacon, OpenDaylight, Floodlight, Maestro, McNettle, Flower, Runos
  - Different programming form Python to Haskell, Erlang
- Для образования - Pox.
- Два больших комьюнити
  - ONOS (Stanford)
  - OpenDayLight (Cisco)
- В России – наш Runos
  - arccn.github.io/runos

Схема работы OpenFlow

Реактивный режим работы

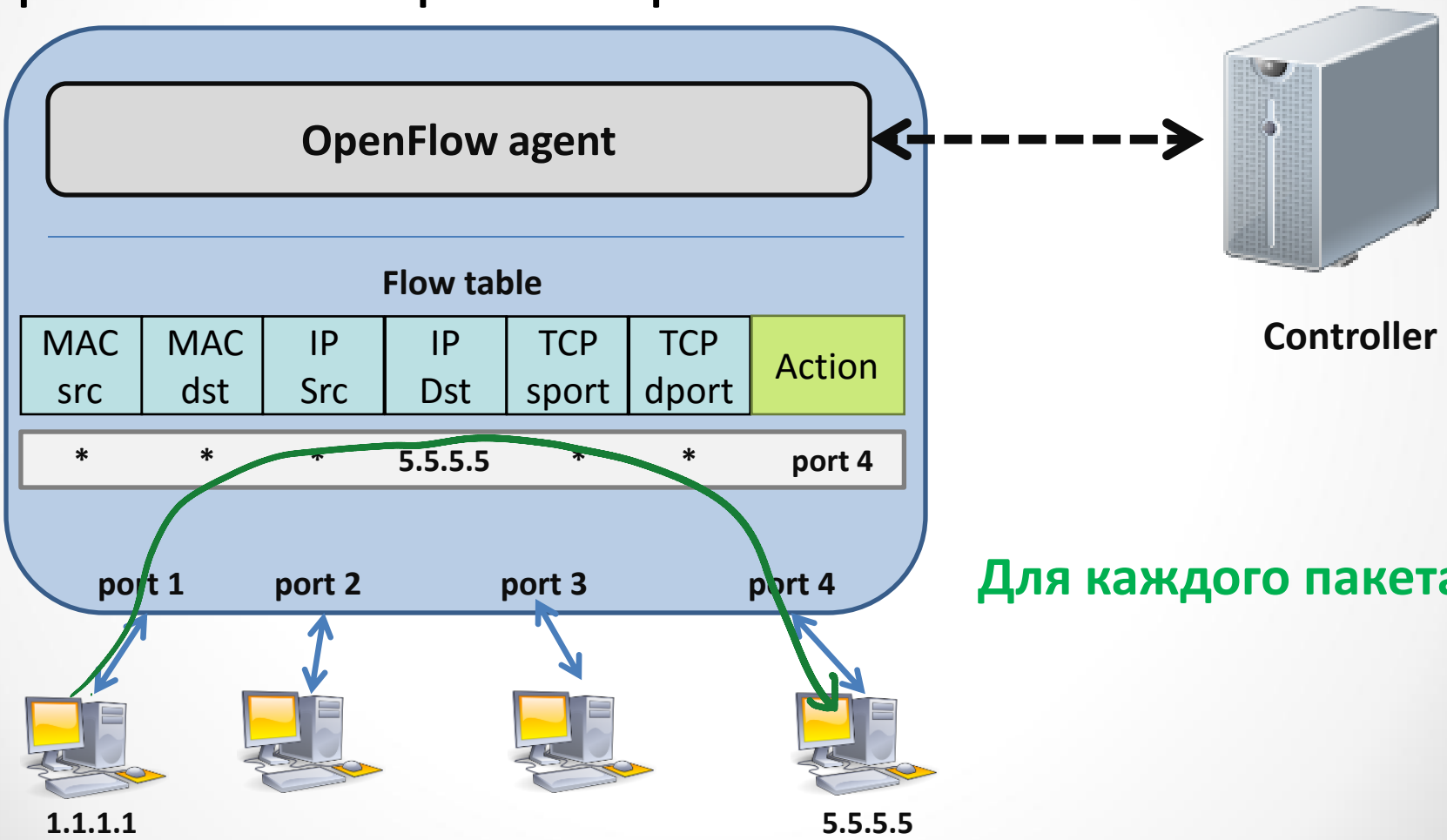


Только для первого пакета

Для всех остальных пакетов

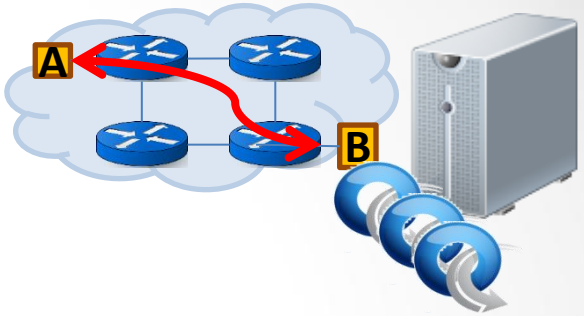
Схема работы OpenFlow

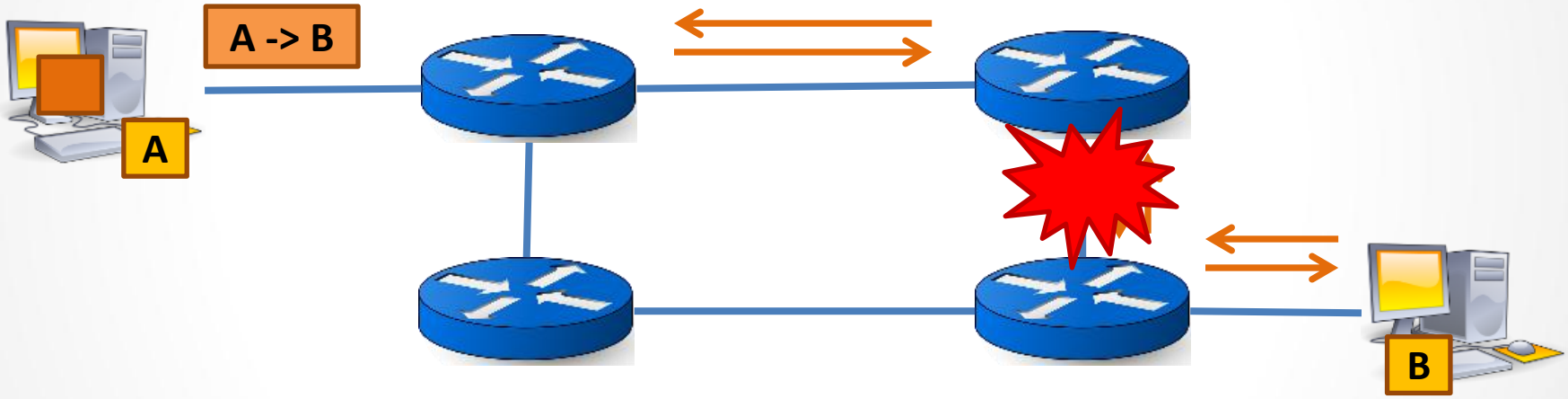
Проактивный режим работы



Для каждого пакета

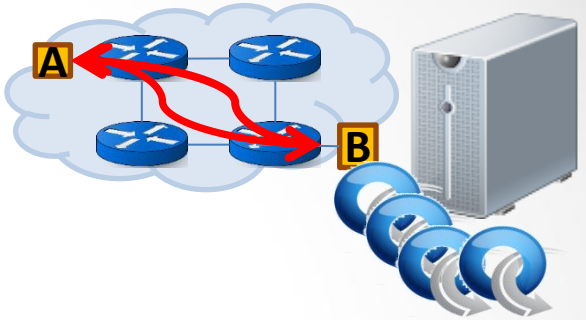
Маршрутизация с SDN/OpenFlow

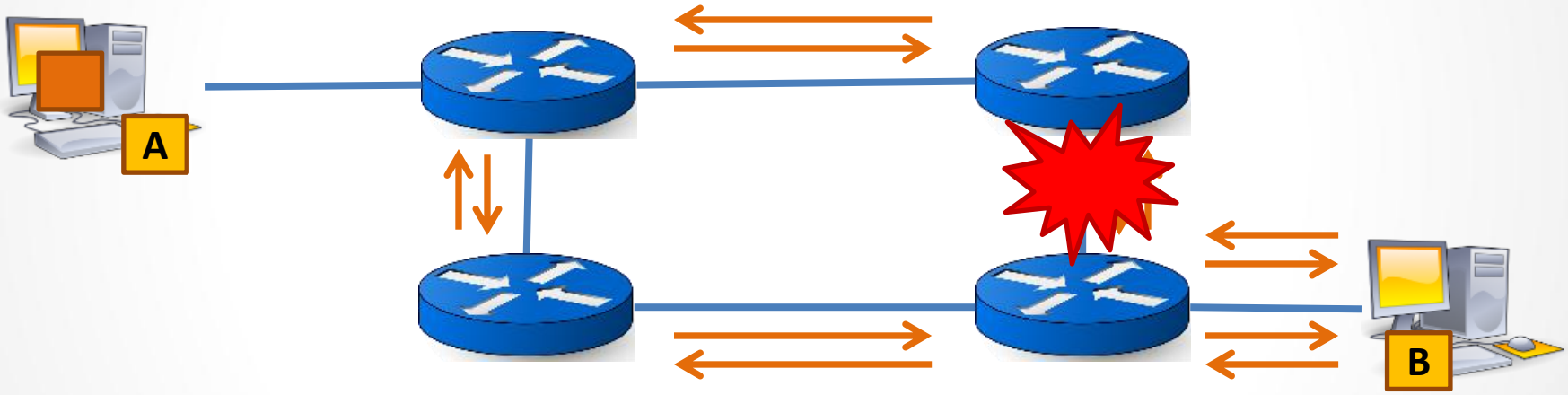




- Неизвестный пакет отправляется на контроллер (OF_PACKET_IN).
- Контроллер вычисляет лучший маршрут через всю сеть (с наименьшей стоимостью и удовлетворяющий политикам маршрутизации).
- Соответствующие правила OpenFlow устанавливаются на коммутаторы + сразу и обратный маршрут (OF_PACKET_OUT/FLOW_MOD).

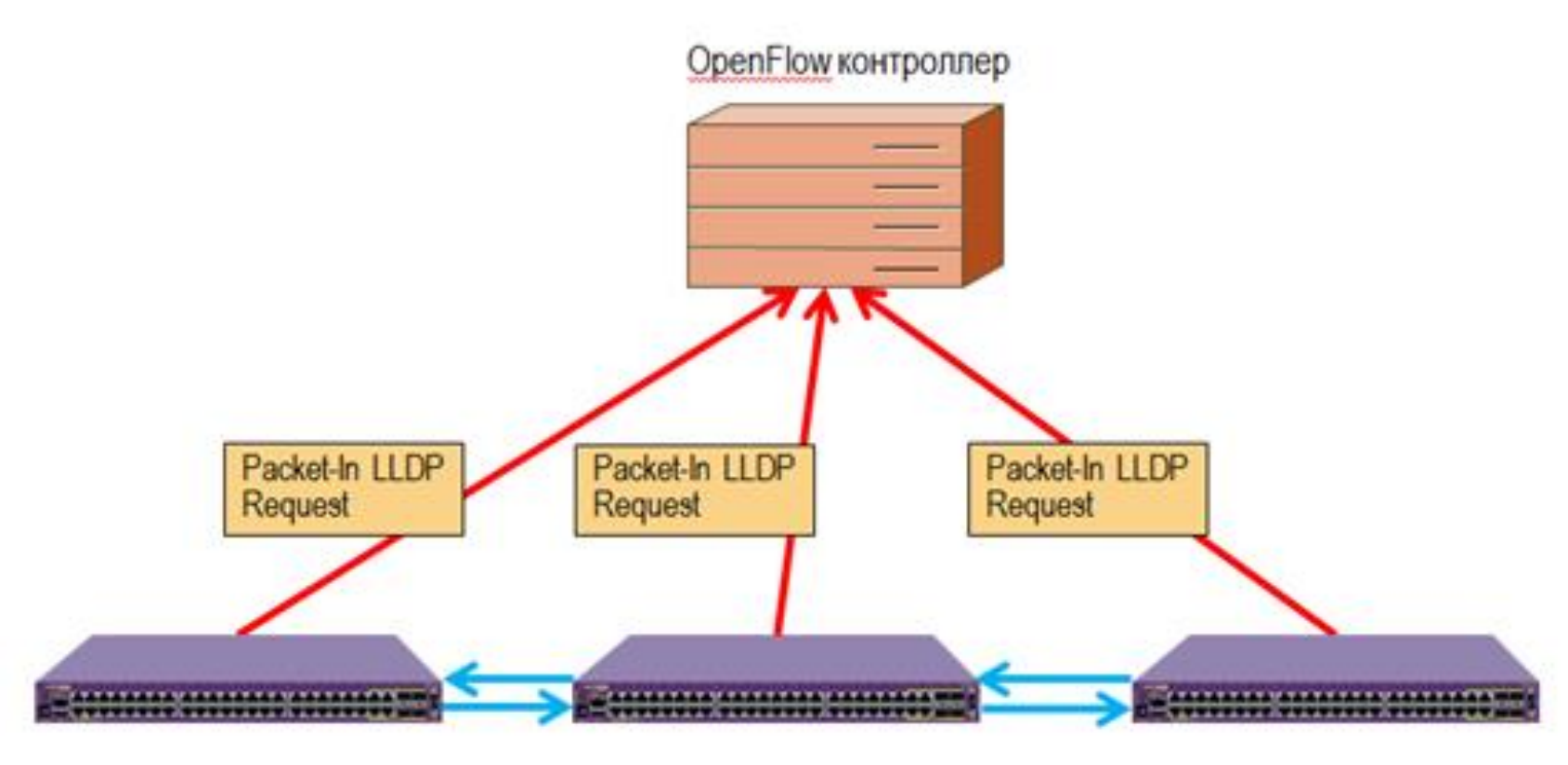
Маршрутизация с SDN/OpenFlow

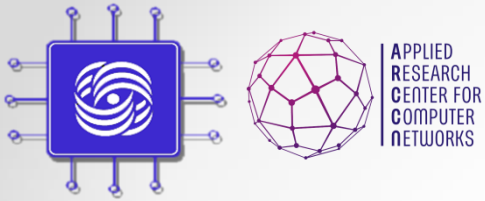




- Неизвестный пакет отправляется на контроллер (OF_PACKET_IN).
- Контроллер вычисляет лучший маршрут через всю сеть (с наименьшей стоимостью и удовлетворяющий политикам маршрутизации).
- Соответствующие правила OpenFlow устанавливаются на коммутаторы + сразу и обратный маршрут (OF_PACKET_OUT/FLOW_MOD).
- **Динамическая переконфигурация в случае ошибки сети.**

Построение топологии?

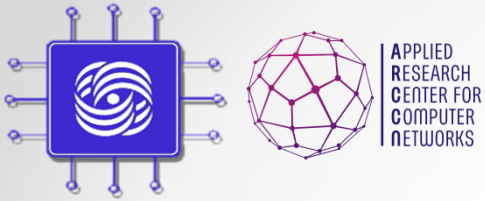




Абстракции OpenFlow

- Уровень управления – forwarding, управление пересылкой пакетов
- Уровень передачи данных – match-action таблицы

Часть III: Варианты применения SDN/OpenFlow

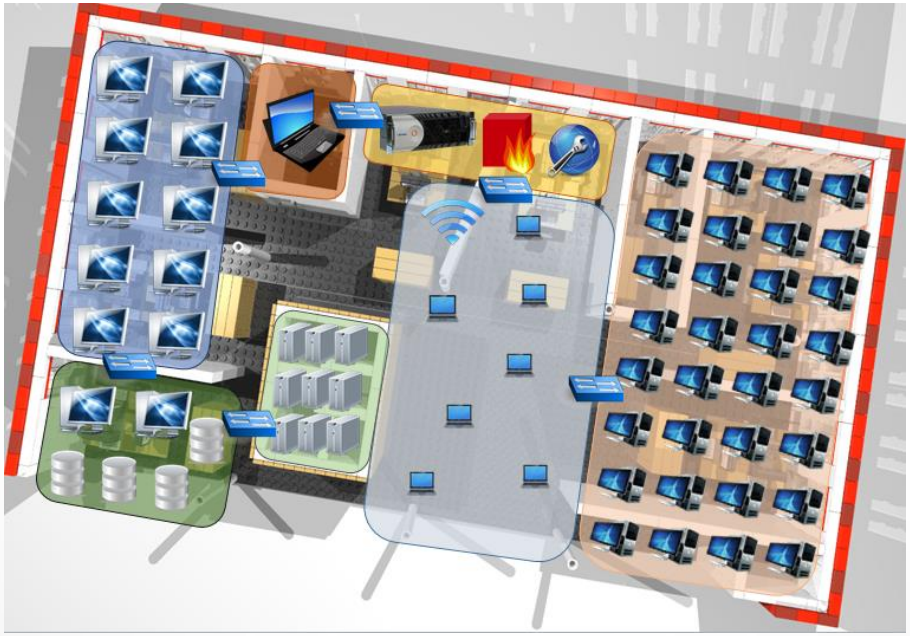


Области применения

- Компании
- Телеком операторы и сервис провайдеры
- ЦОД и облака

Корпоративная сеть

- Современные компания имеют сложную сетевую инфраструктуру:
  - Большое количество сетевых элементов
  - Разветвленная топология
  - Набор различных политик маршрутизации и безопасности



Трудности администрирования

- Сетевые администраторы отвечают за поддержания работы сетевой инфраструктуры:
  - Сетевые инженеры руками переводят высокоуровневые политики в низкоуровневые команды
  - Ручная настройка всех сетевых устройств
  - Ограниченный инструментарий по управлению сетевыми устройствами
  - Переучивание под каждого вендора

Существующие системы управления

- Предназначены для мониторинга состояния: топология, характеристики каналов, загрузка каналов и задержка.
- Основы на протоколе SNMP.
- Конфигурация оборудования по-прежнему происходит в ручном режиме.

Примеры: Cisco Prime, HP OpenView, IBM Tivoli, OpenNMS.

Цель

1. Сделать сеть управляемой без ручного доступа к оборудования.
2. Повысить уровень абстракции управления сетью.

Семантическое управление сетью

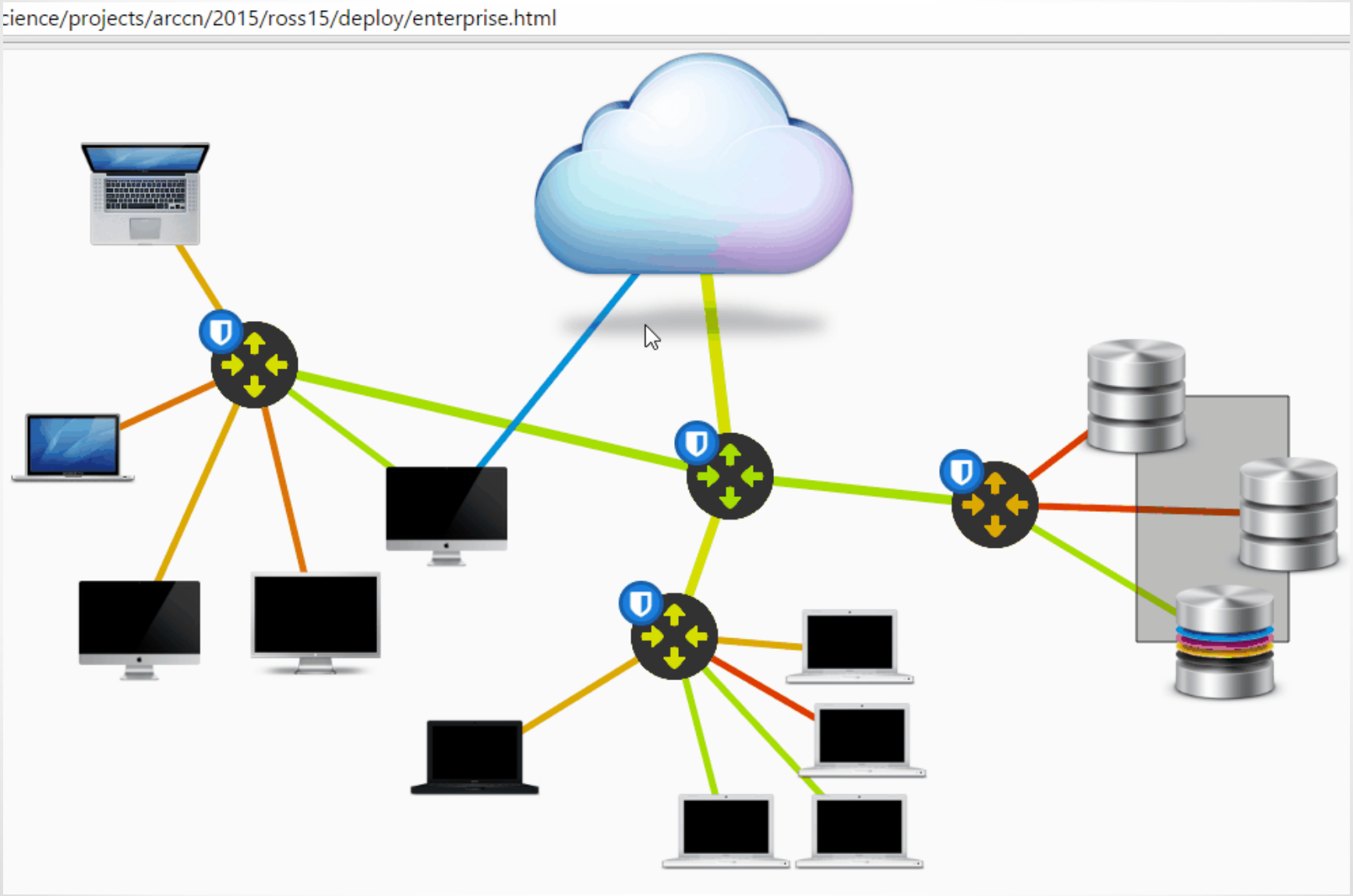
- **Имена**. Работа с имена хостов проще, чем запоминать связки IP и MAC адресов.
- **Группы**. Хосты объединяются в группы, удобно для задания одних политик. Вместо подсетей.
- **Пути**. Задавать маршруты через всю сеть сразу, а не для каждого устройства отдельно.

Бонус: В связи с тем, что теперь IP адреса скрыты, выбираем их сами, так как нам это удобно. Например, уменьшение правил, агрегирование потоков.

Основные функции

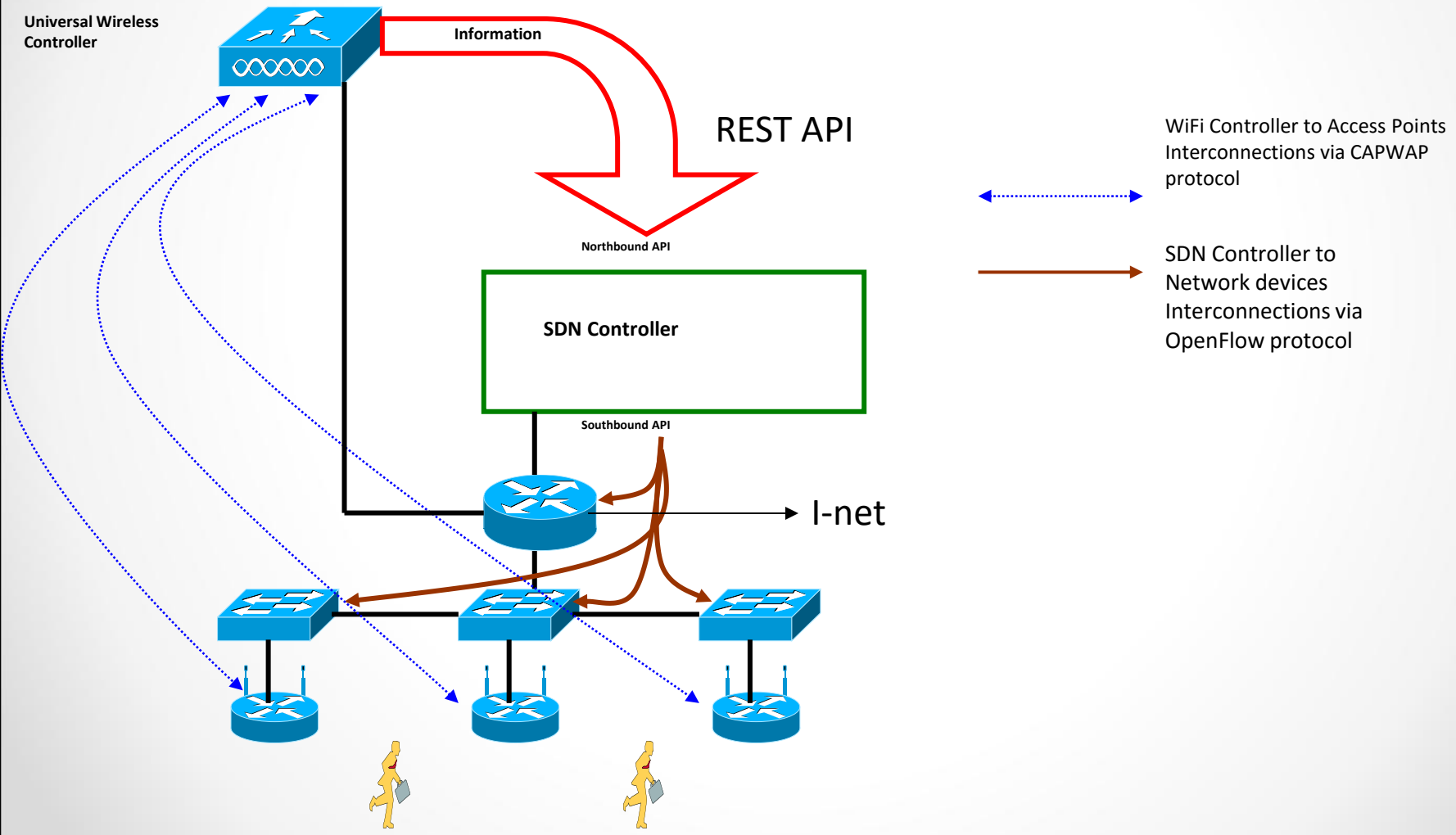
- Автоматическое определение топологии.
- Мониторинг загрузки сети (пропускная способность, задержки).
- Именование.
- Группы.
- Выбор маршрутов.
- Поддежка QoS.
- Балансировка нагрузки.
- Файрволл и ограничения доступа.
- NAT

Как это выглядит?



WiFi Controller & SDN Networks

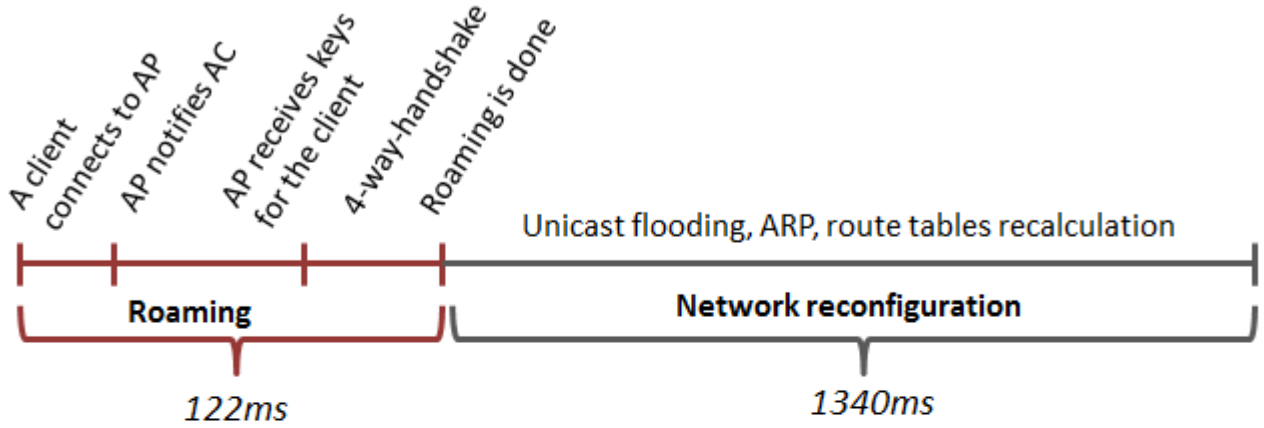
The principle of interaction with SDN controller over Northbound API:

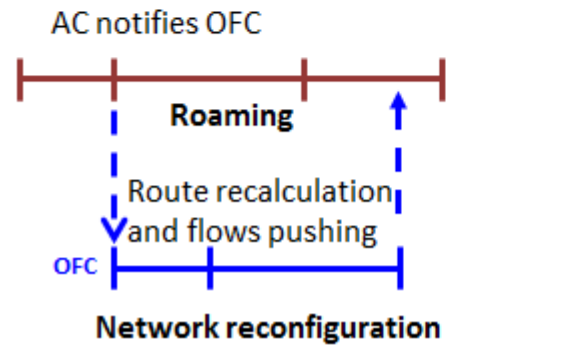


WiFi Controller & SDN Networks

Test bed consists of:

- Three hardware OpenFlow switches Extreme Networks SummitX 460t
- Two TP-LINK access points
- A laptop moving from one AP to another one and running ping command to the outside server
- Both controllers run on the same server



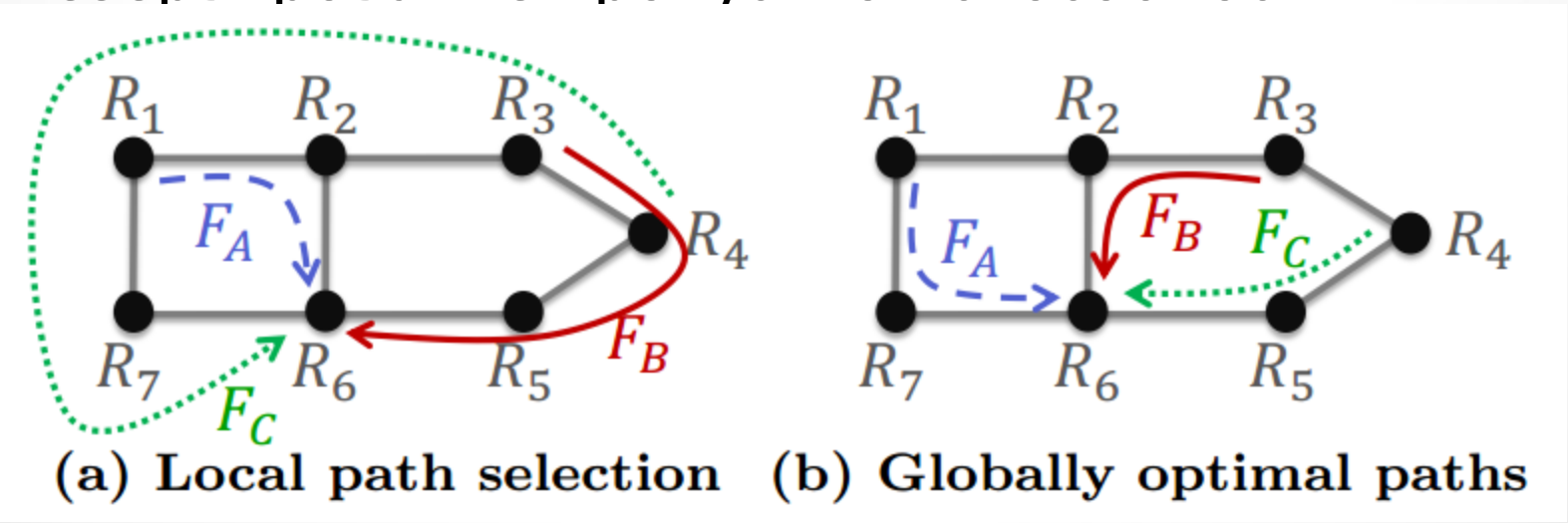


- The legacy network needs in average 1.5 seconds to reconfigure, while the SDN/OpenFlow network doesn't bring additional delay.
- This is because the migration procedure in Chandelle requires less than 80ms and the OpenFlow controller has enough time to reconfigure the switches.
- **Finally, we have more faster roaming with SDN/OpenFlow.**

Телеком

1. Интеллектуальный Traffic Engineering:

- Выбор оптимального пути
- Реакция на отказ канала
- Резервирование пропускной способности



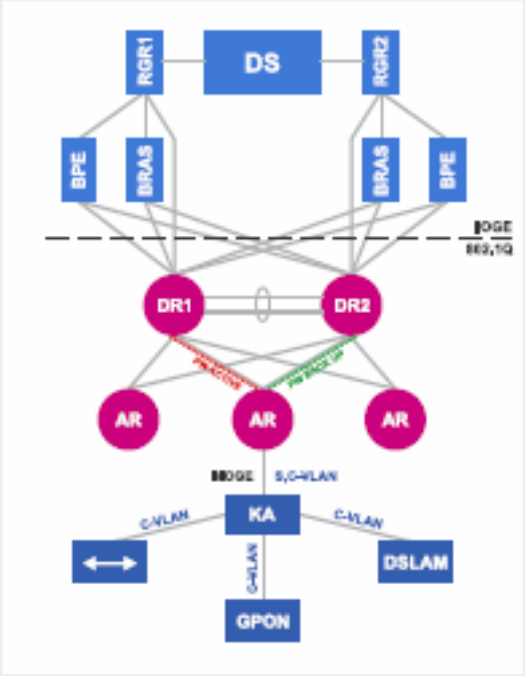
Телеком

- Как применить все это на практике?
  - Greenfield?
  - Проблемы интеграции с традиционной сетью
    - Нужно подыгрывать протоколам традиционной сети, т.е. правильно отвечать на запросы.
    - Чем меньше стыков с традиционной сетью, тем лучше.
  - Проблема интеграции с существующими системами управления

WAN segment (Service Provide)

Services:

- L2 transit for B2C, B2B/G (Internet, VPN)
- Fast backup path
- SLA
- IPTV multicast
- VoIP
- Mobile backhaul



Before:

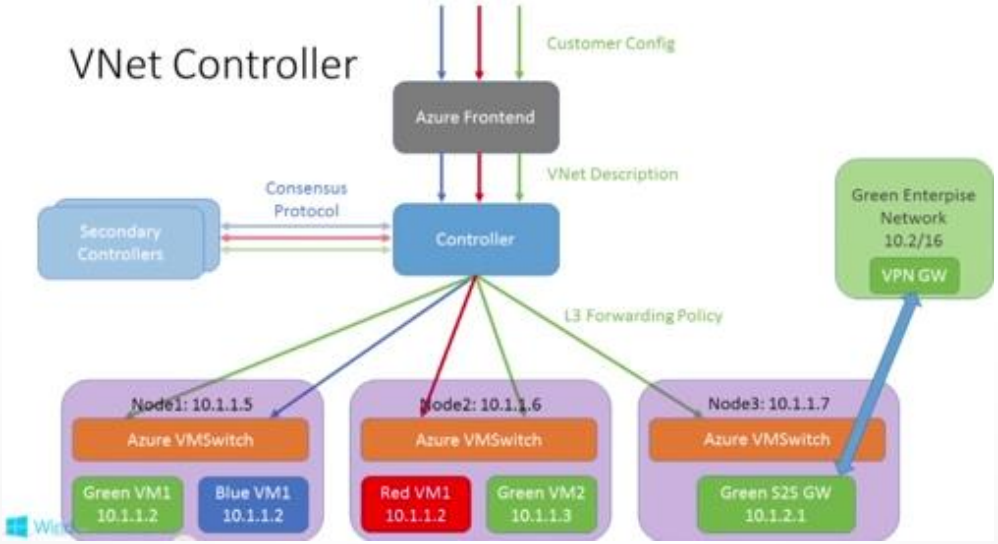
- IS-IS (RFC 1195)
- OSPF
- PIM-SSM
- PIM-SM
- LDP (RFC 3036)
- Targeted LDP
- BGP (PW для AToM)
- RSVP (RFC 2205)
- MPLS PW
- BGP (RFC 4271)
- MP-BGP (RFC 4760)
- MPLS-VPN (RFC 4364)
- MPLS (RFC 3031, 3032)

After:

- L2 PW application + stats (no encap)
- Bridge domain (no learning)
- Multicast (IGMP)
- L2 LAG, L3 ECMP
- H-QoS

ЦОД/Облака

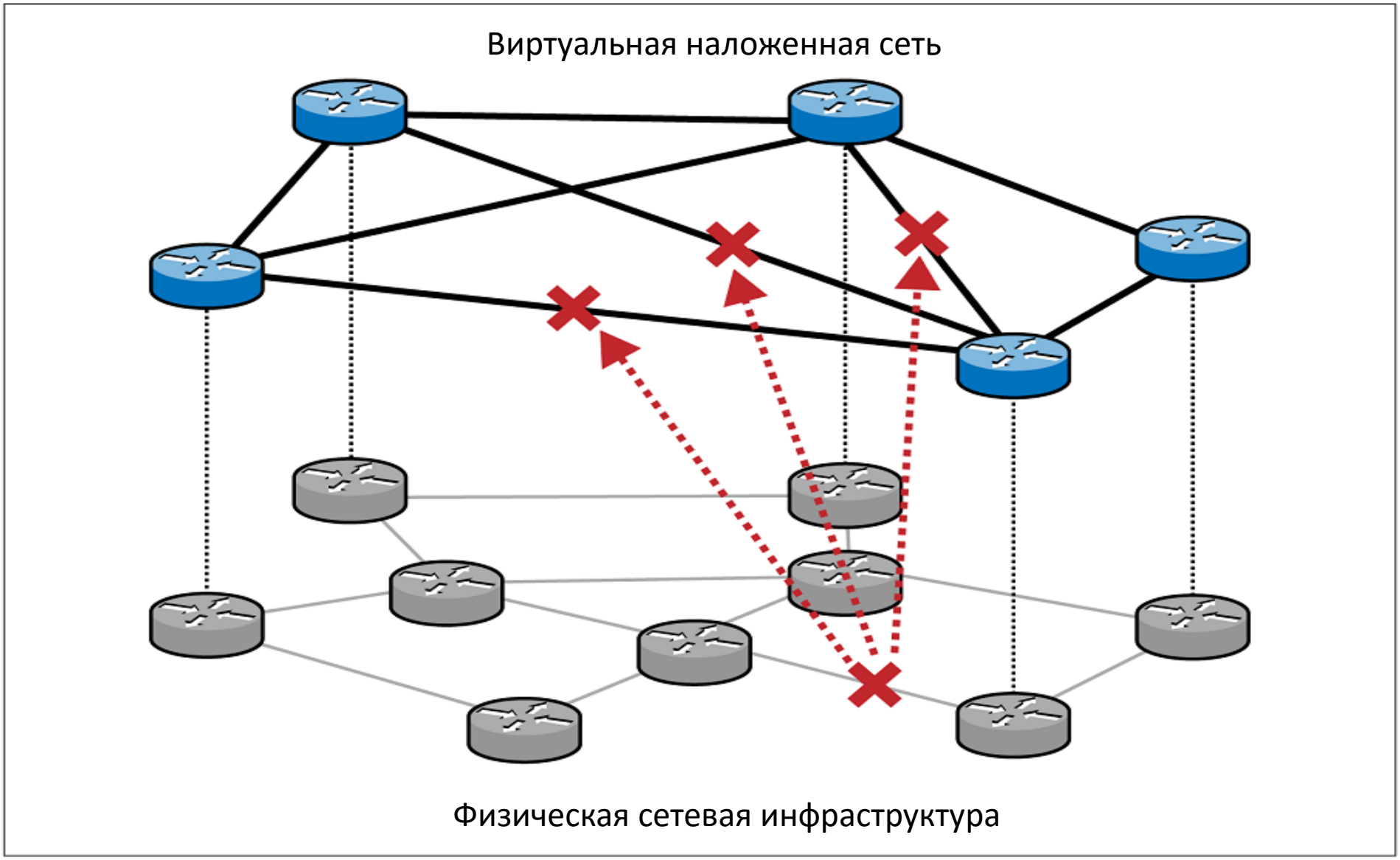
- Повышение утилизации оборудования и каналов
- Мониторинг и оптимизация потоков
- Виртуализация сети пользователей
- Балансировка нагрузки
- Обеспечение качества доступа



ЦОД/Облака

- Как правило есть два SDN
  - Без OpenFlow и так есть в OpenStack
    - ТОЛЬКО виртуальные каналы
    - Туннели, таблицы, новые VM, политики
  - С OpenFlow для управление физическими устройствами
    - Качество канала, определение узких мест

Сетевая виртуализация



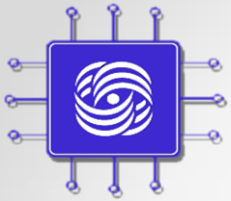
Software Defined Data Center

Плюсы

- Оптимизация управления инфраструктурой ЦОД
- Уменьшение зависимости от аппаратуры
- Автоматизация выполняемых задач
- Масштабируемость, эффективность, гибкость
- Быстрота реализации требуемого функционала

Минусы

- Разнородность решений разных производителей
- Отсутствие зрелых стандартов
- Сырость предлагаемых решений
- Недостаточный набор реализованных функций



Quiz 1

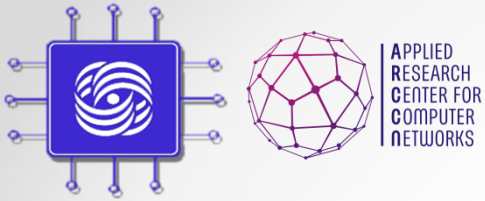
- На другом слайде

Заключение

- SDN уже активно используется в промышленности и является основным трендом в развитии телеком индустрии.
- SDN != OpenFlow
  - SDN – подход к разделению уровня данных и уровня управления
  - OpenFlow – одна из реализаций. Другие, XMPP, SNMP, overlay.

"SDN means thinking differently about networking"

http://arccn.ru/ ashalimov@lvk.cs.msu.su @alex_shali @arccnnews



Видео об SDN

- Немного юмора
  - SDN с разных точек зрения
  - http://www.youtube.com/watch?v=GRVygzcXrM0