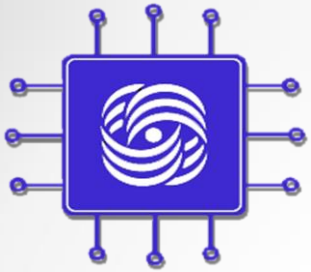


APPLIED
RESEARCH
CENTER FOR
COMPUTER
NETWORKS

Технологии SDN/OpenFlow

Математический спецкурс
“Программно Конфигурируемые Сети”

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

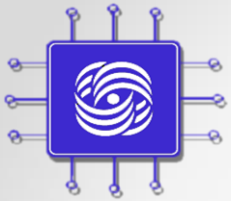
Google перевел сеть между ЦОД на SDN в 2012 году, сейчас анонсирована внутренняя облачная платформа **Andromeda**.

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”*.

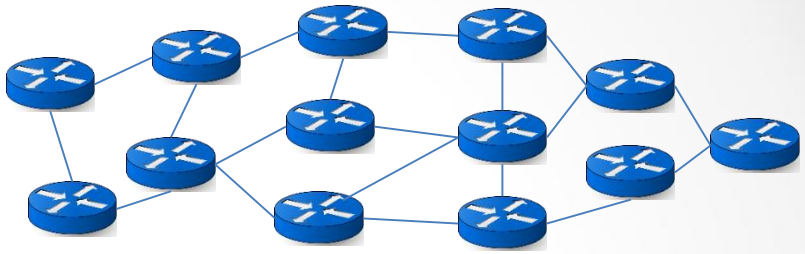


А что с SDN в России?

Ростелеком

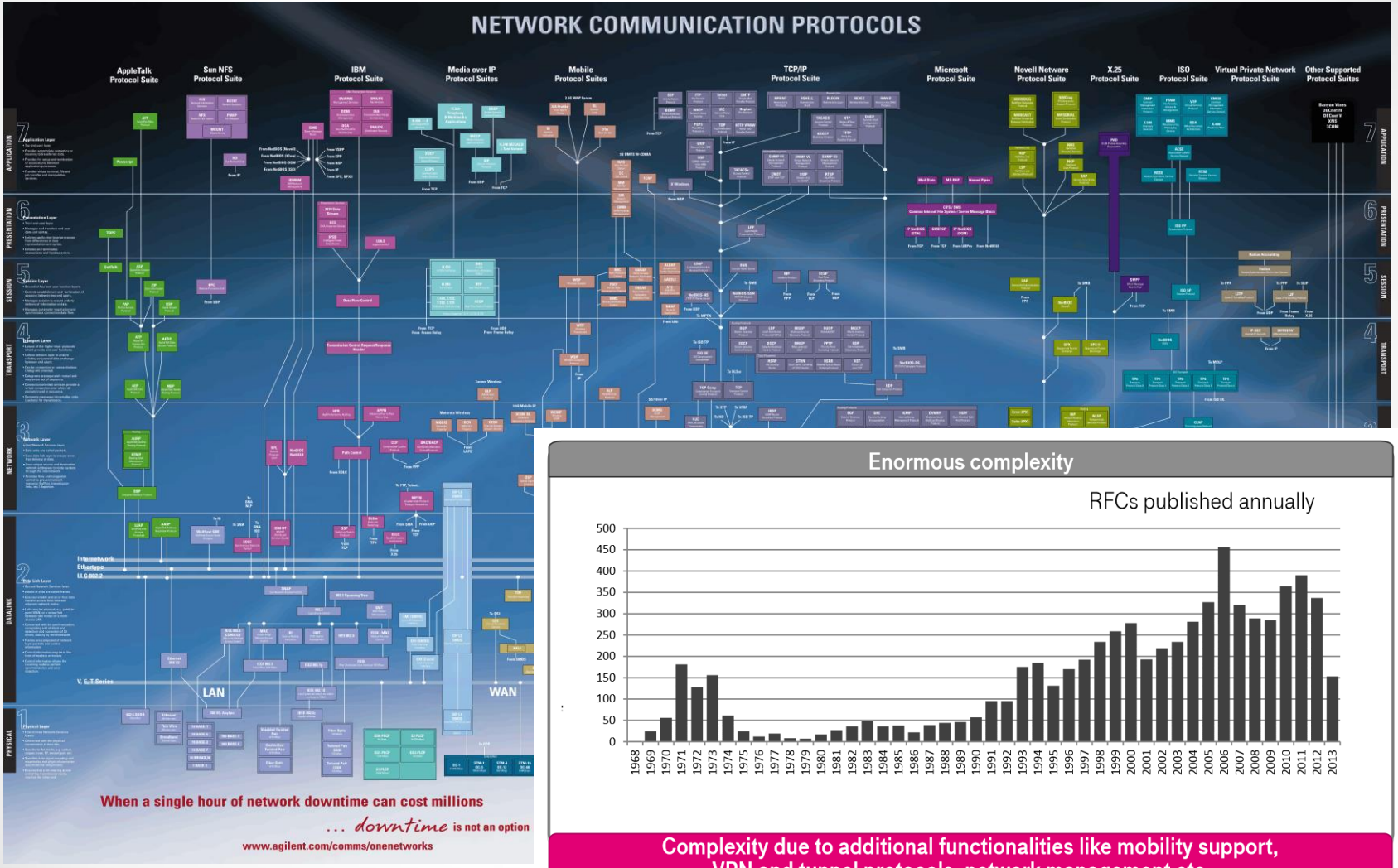
- «Ростелеком» начал работу над внедрением перспективных технологических направлений **«Программно-конфигурируемых сетей» (SDN)** и **«виртуализации сетевых функций» (NFV)**.
- «Ростелеком» и впредь намерен укреплять **технологическое лидерство**. Новые технологии позволят **упростить сетевую инфраструктуру** 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/

Что такое SDN/OpenFlow?

SDN = Software Defined Networking

Внедрения

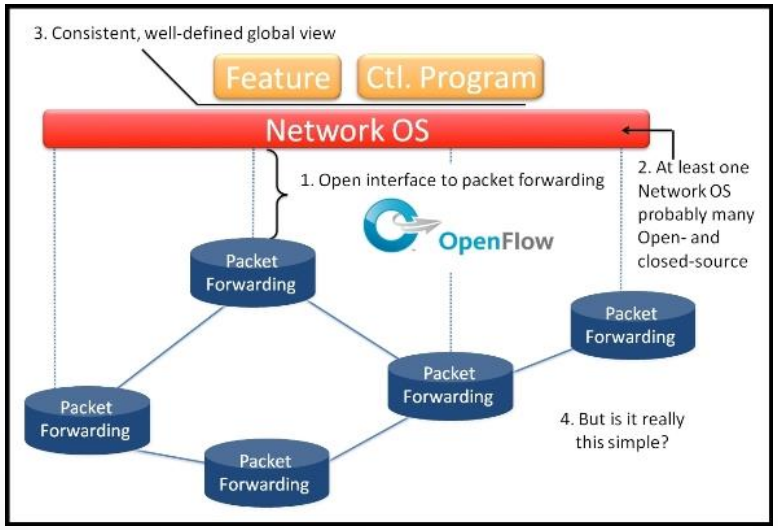
Google

NTT Communications

- •
- •
- •

Основные принципы

- Физическое разделение уровня передачи данных от уровня управления сетевых устройств.
- Логически централизованное управление.
- Программируемость.
- Открытый единый интерфейс управления.

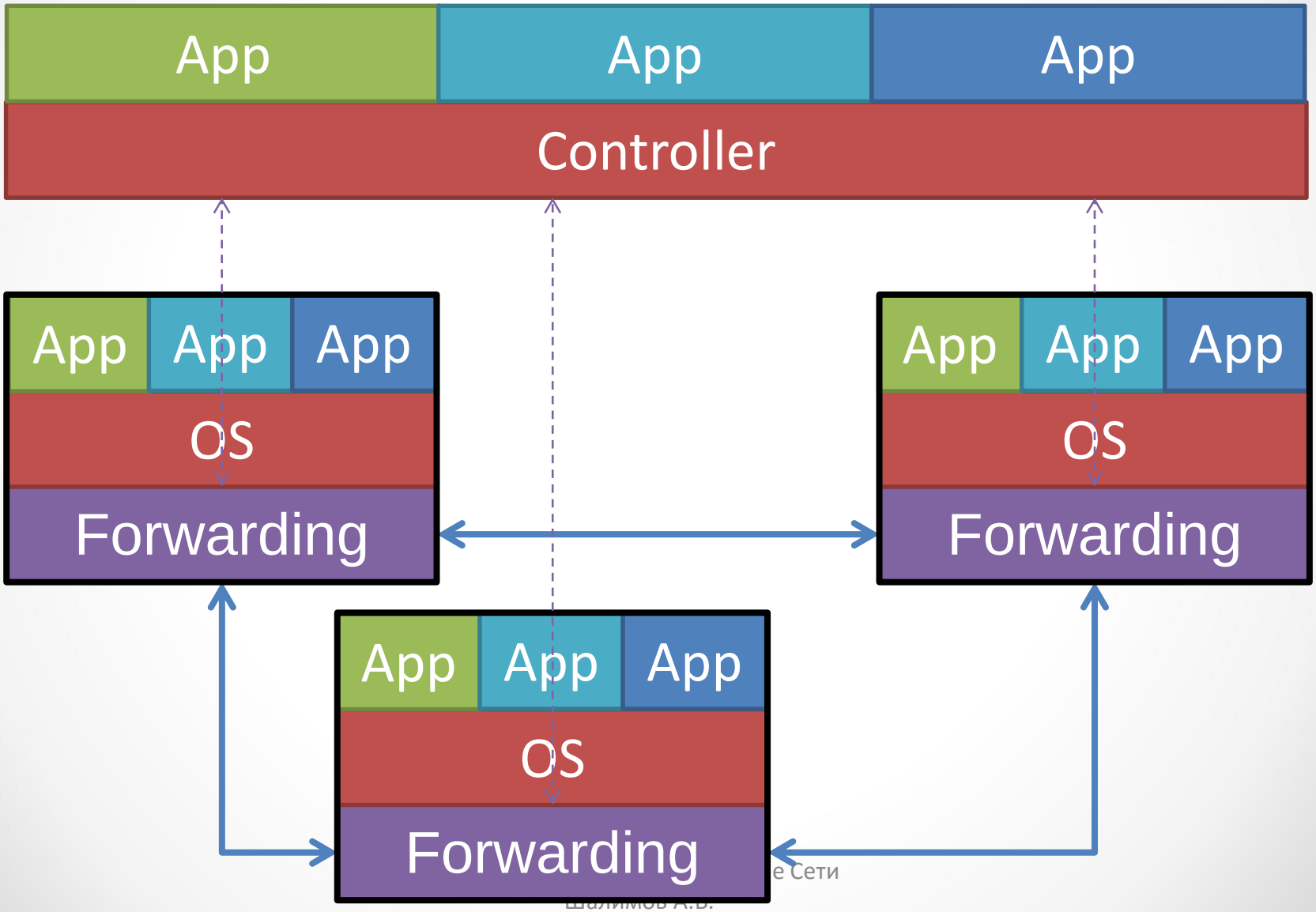


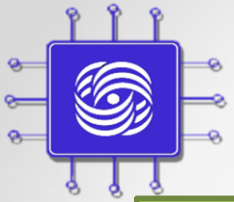
Преимущества

- Упрощение управления сетью (OPEX)
- Удешевление оборудования (CAPEX)
- Разработка ранее недоступных сервисов

"SDN means thinking differently about networking"

Переход к SDN





Архитектура SDN

App

App

App

Controller

Control Protocol

Usable GUI

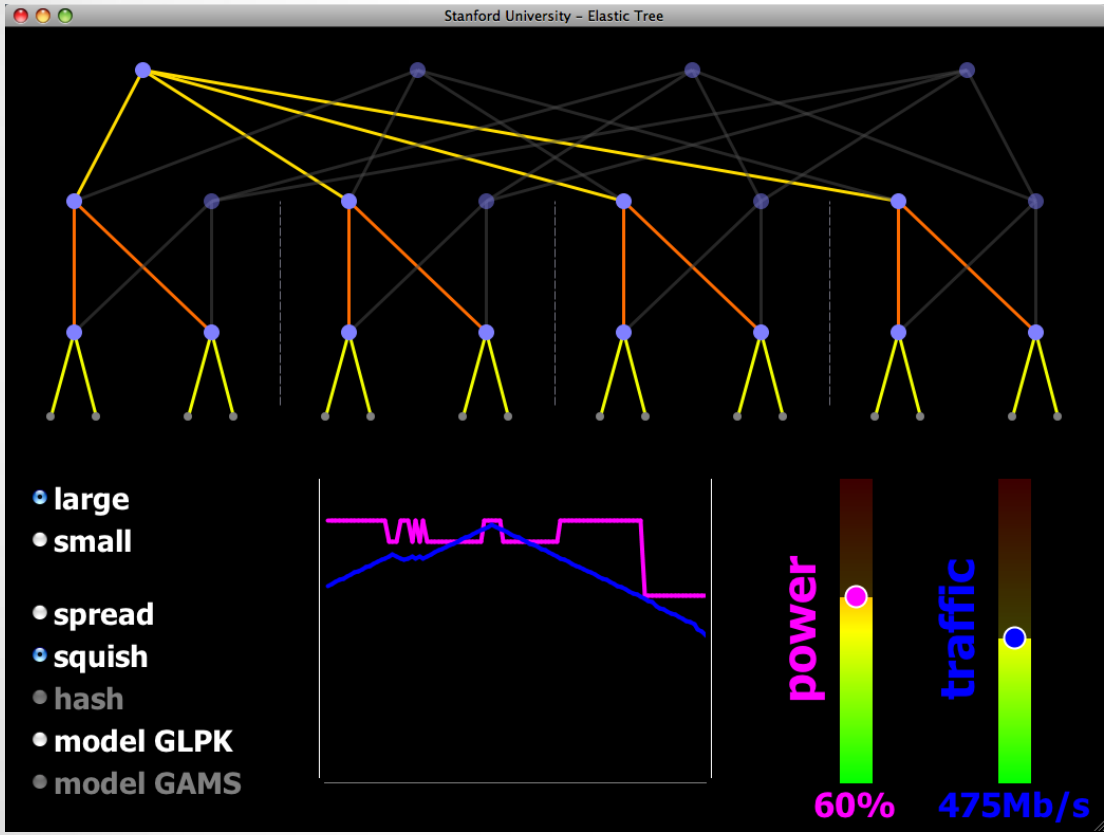
OS Services

Switch

Switch

Switch

Пример применения

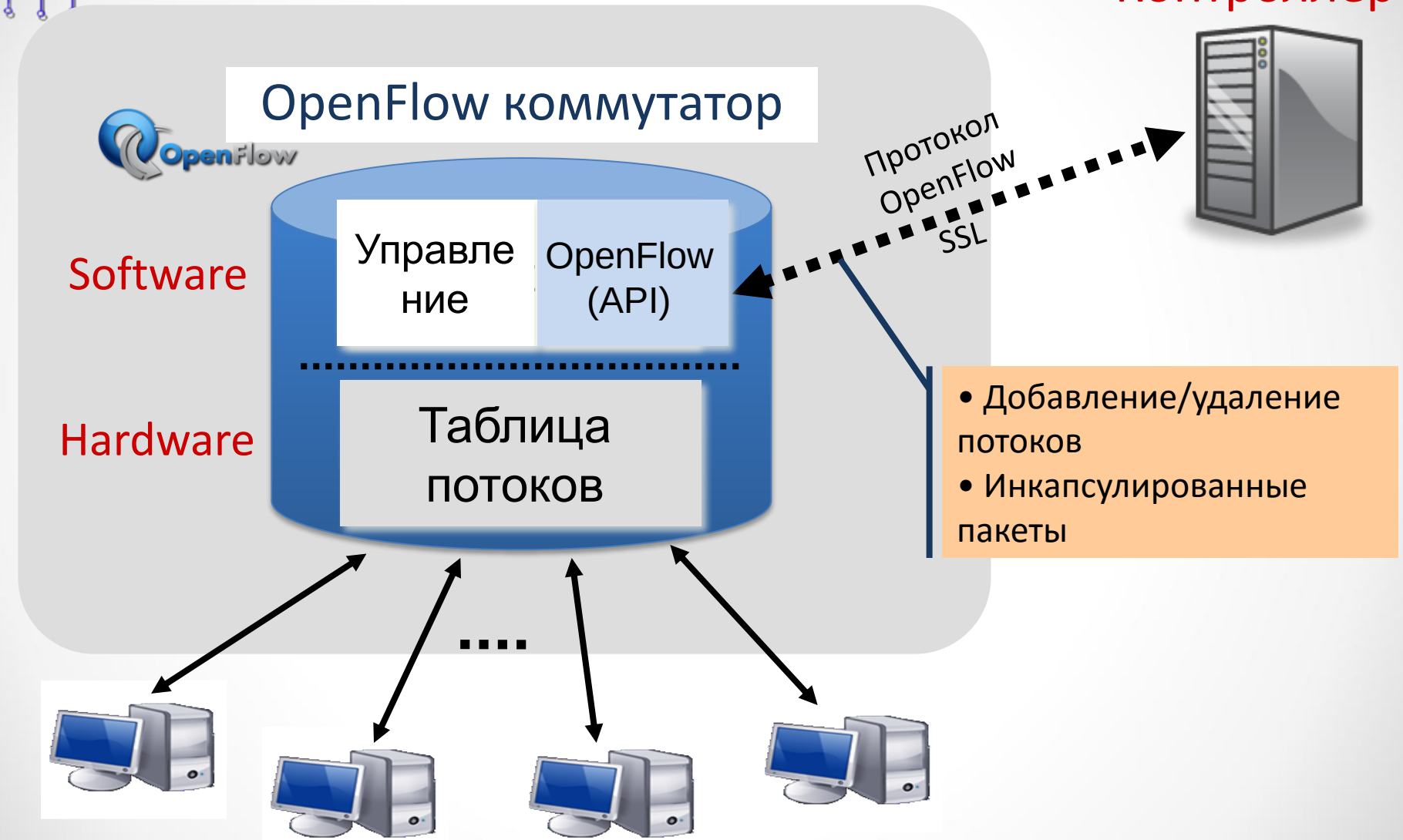


Уменьшение энергопотребления в ЦОД

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

Часть II: OpenFlow

OpenFlow


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

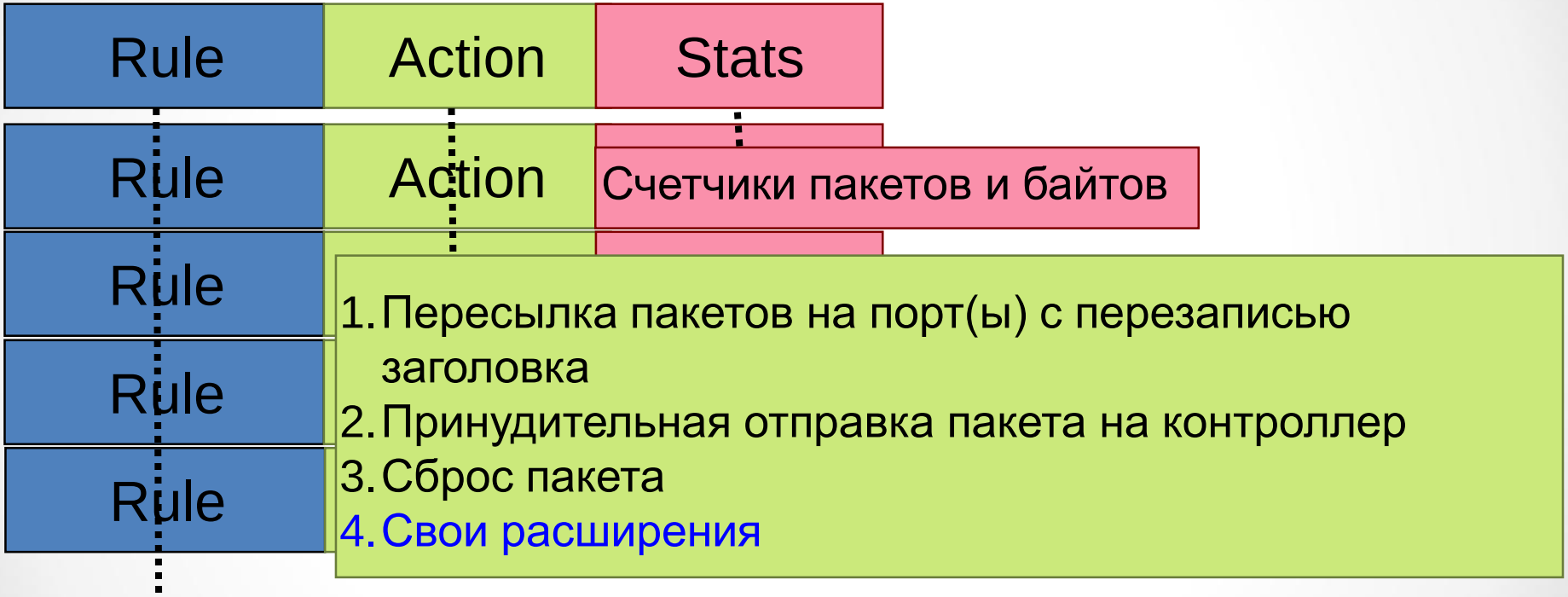
OpenFlow протокол

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

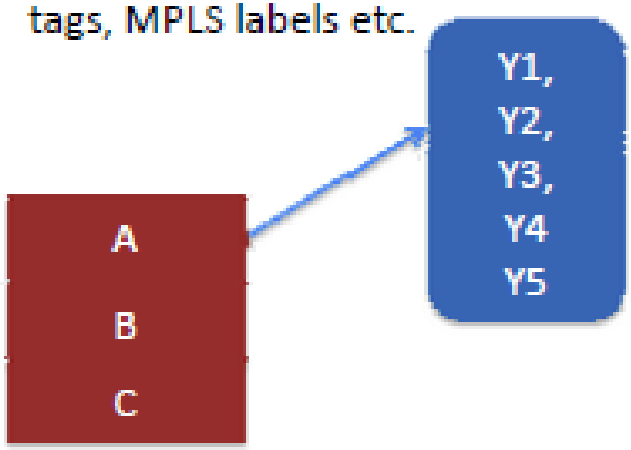
Firewall

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.

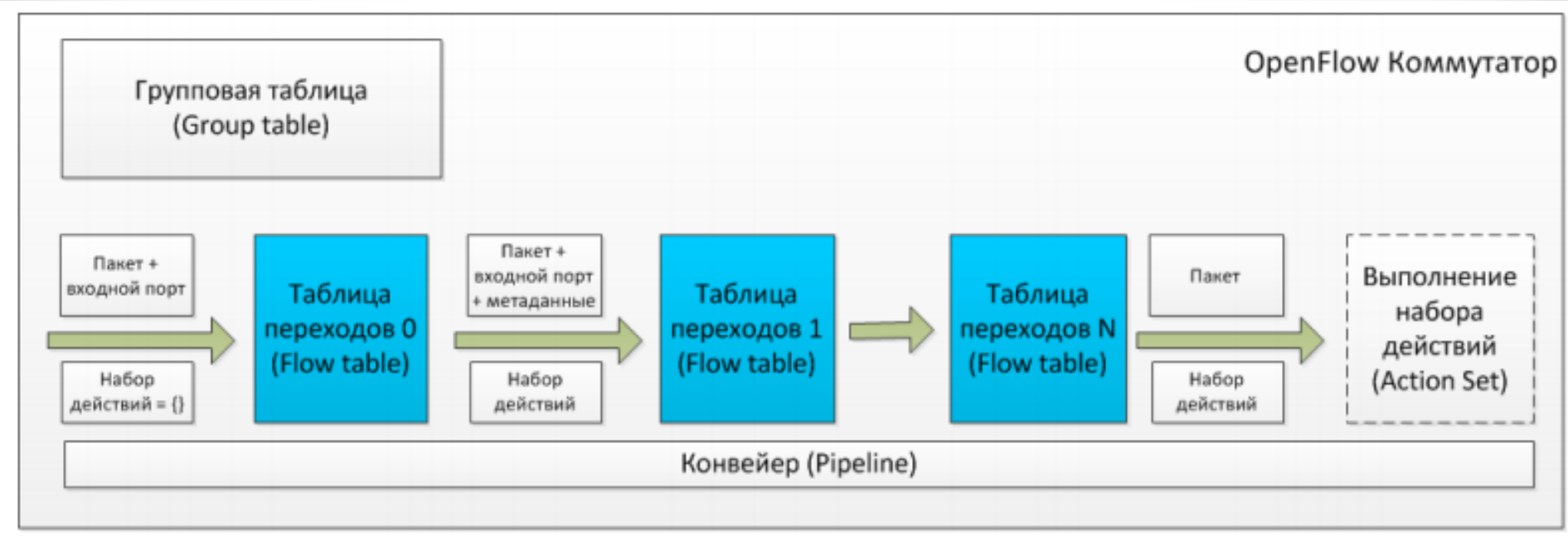


OF 1.0 Single Table

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

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

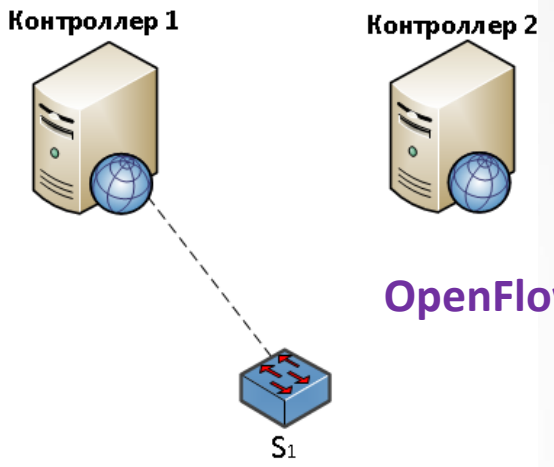
- Для реализации QoS и ограничения скорости
  - Для каждого потока или группы потоков
  - Следит за превышение значений счетчиков
  - Действия: **drop** или **dscp remark**

Meter Identifier	Meter Bands	Counters

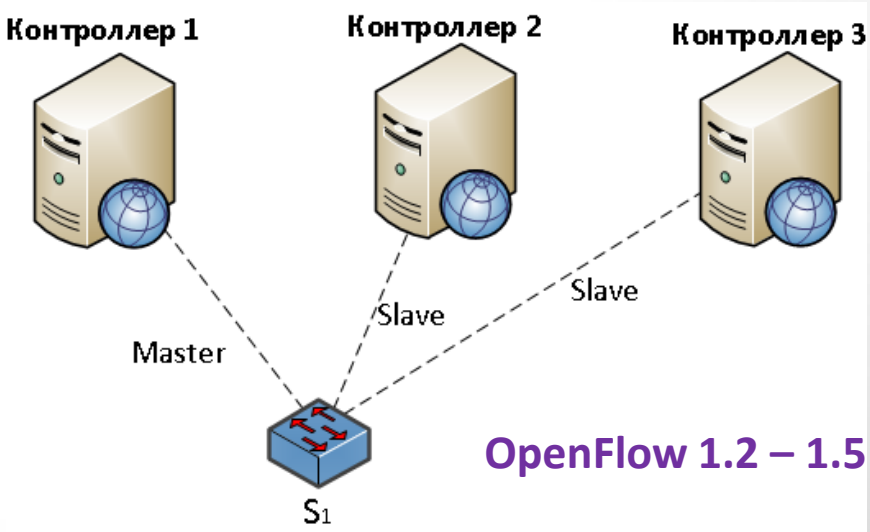
Band Type	Rate	Counters	Type specific arguments

Несколько контроллеров

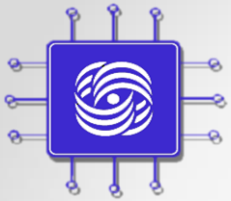
- Протокол OpenFlow 1.2:
  - Множество контроллеров
  - Механизм ролей
  - **Роли:** Master, Slave, Equal
  - **По умолчанию:** контроллер находится в роли Equal для коммутаторов.
  - **Смена роли:** OFPT_ROLE_REQUEST
  - **Распределение ролей:** возложено на контроллеры.



OpenFlow 1.0, 1.1



OpenFlow 1.2 – 1.5




OpenFlow контроллер

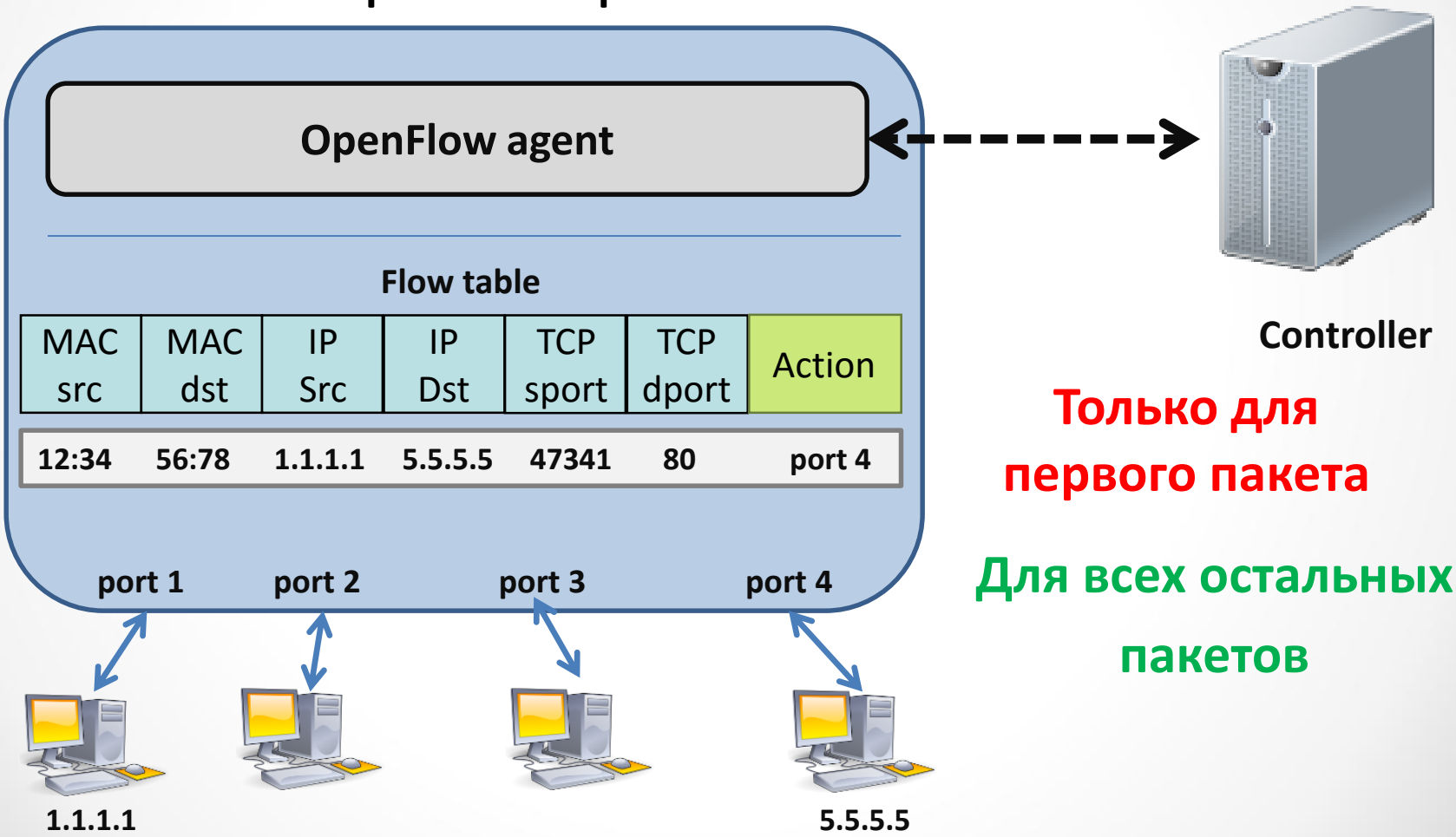
- Программа, TCP/IP сервер, ожидающий подключения коммутаторов
- Отвечает за обеспечение взаимодействие приложения-коммутатор.
- Предоставляет важные сервисы (например, построение топологии, мониторинг хостов)
- API сетевой ОС или контроллер предоставляет возможность создавать приложения на основе **централизованной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

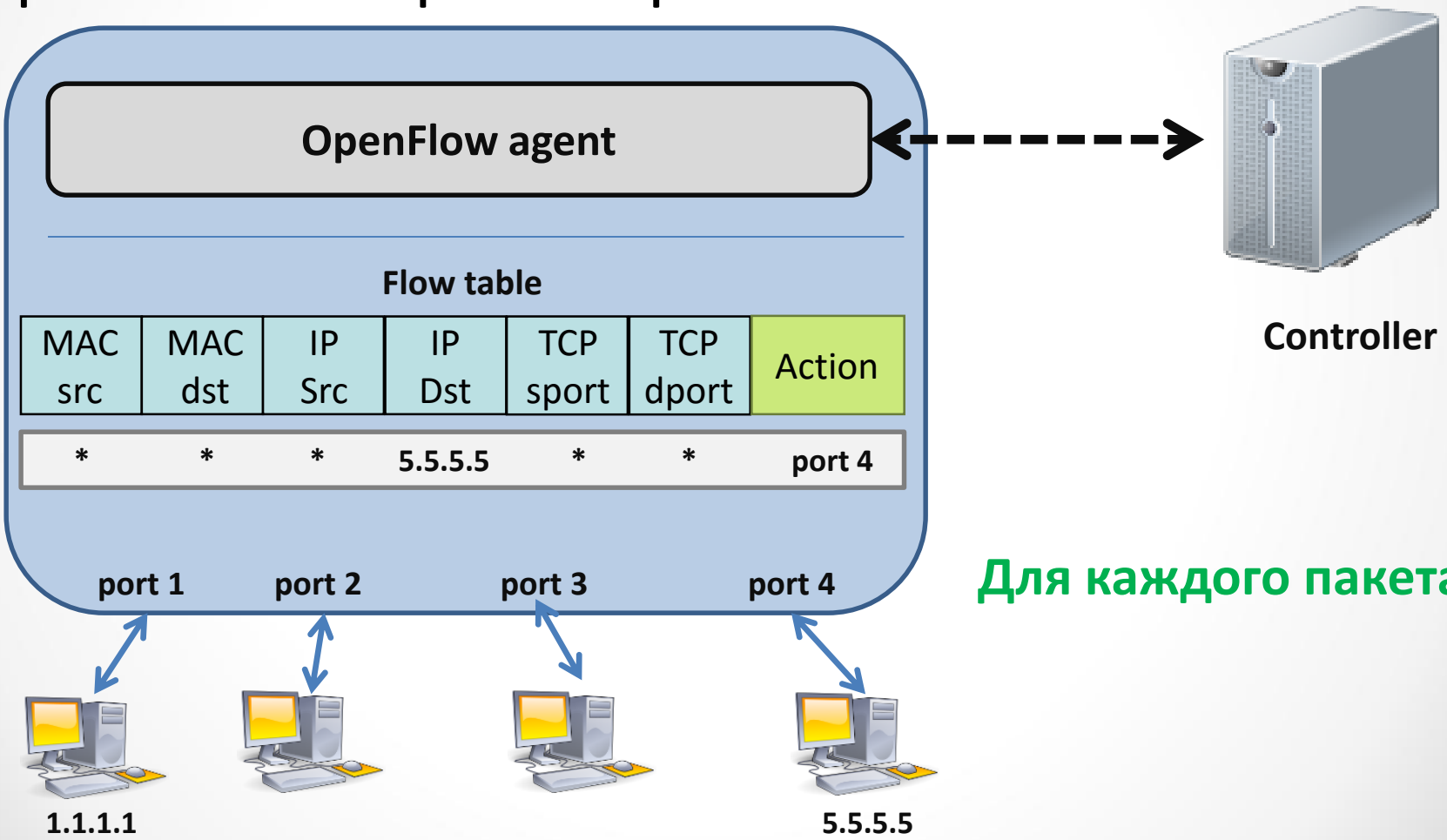


Только для первого пакета

Для всех остальных пакетов

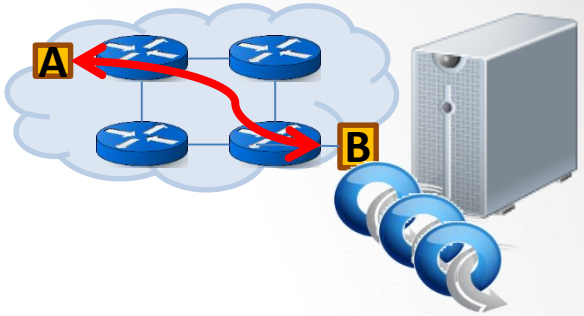
Схема работы OpenFlow

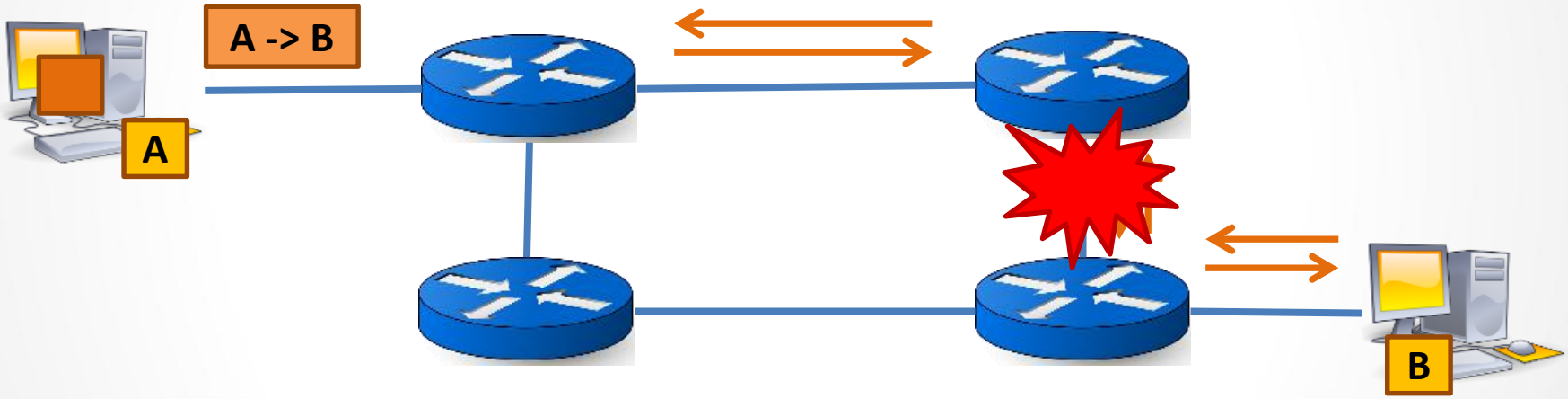
Проактивный режим работы



Для каждого пакета

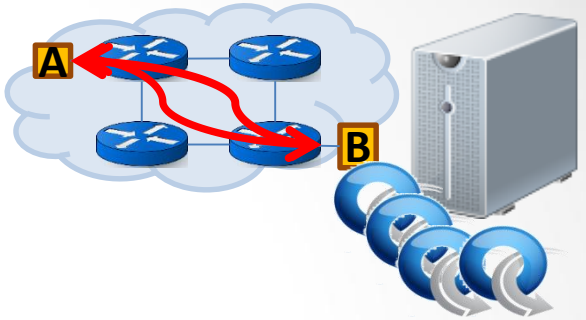
Маршрутизация с SDN/OpenFlow

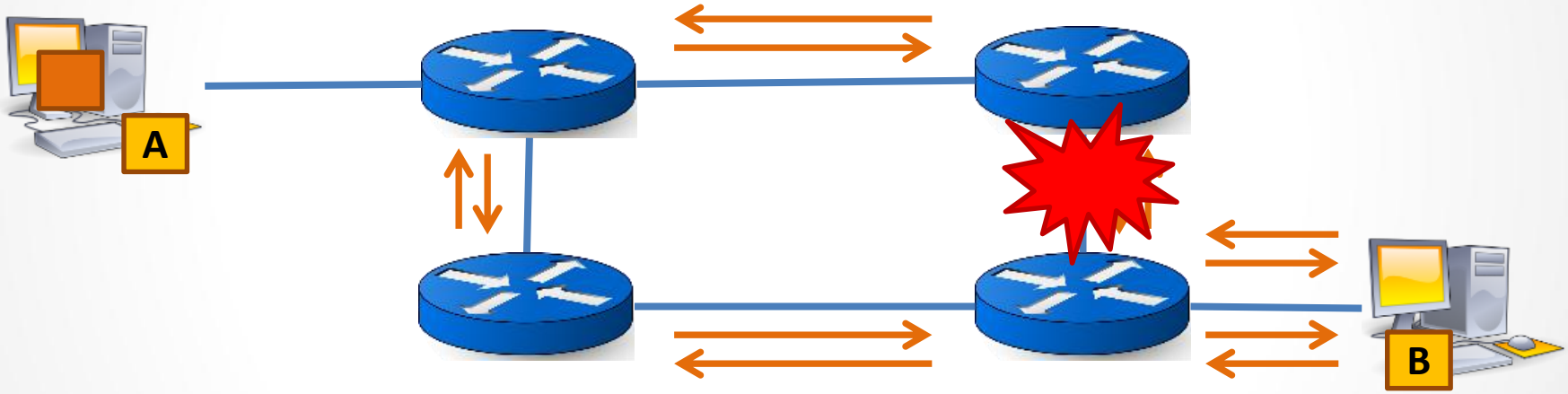




- Неизвестный пакет отправляется на контроллер (OF_PACKET_IN).
- Контроллер вычисляет лучший маршрут через всю сеть (с наименьшей стоимостью и удовлетворяющий политикам маршрутизации).
- Соответствующие правила OpenFlow устанавливаются на коммутаторы + сразу и обратный маршрут (OF_PACKET_OUT/FLOW_MOD).

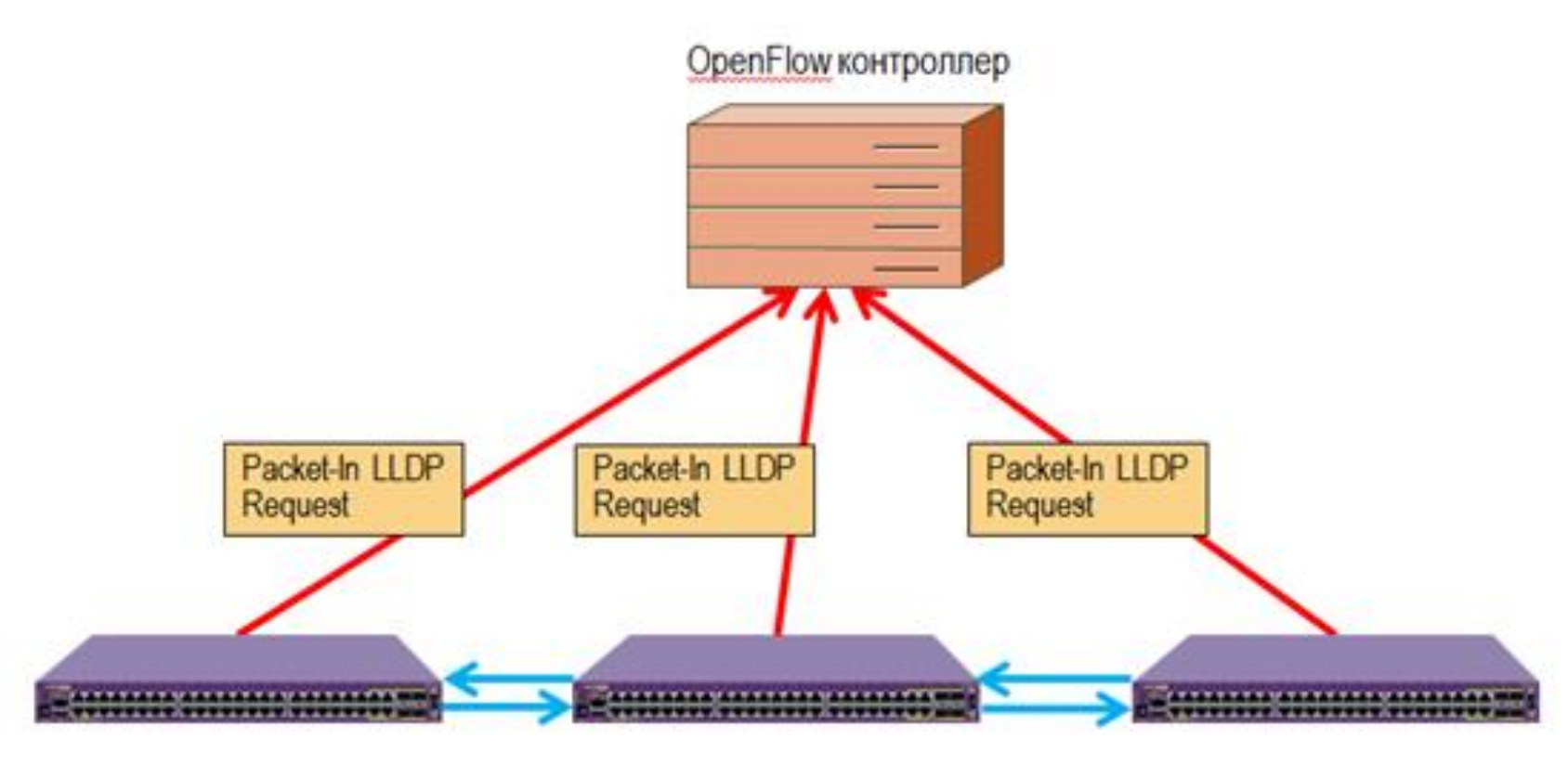
Маршрутизация с SDN/OpenFlow





- 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?

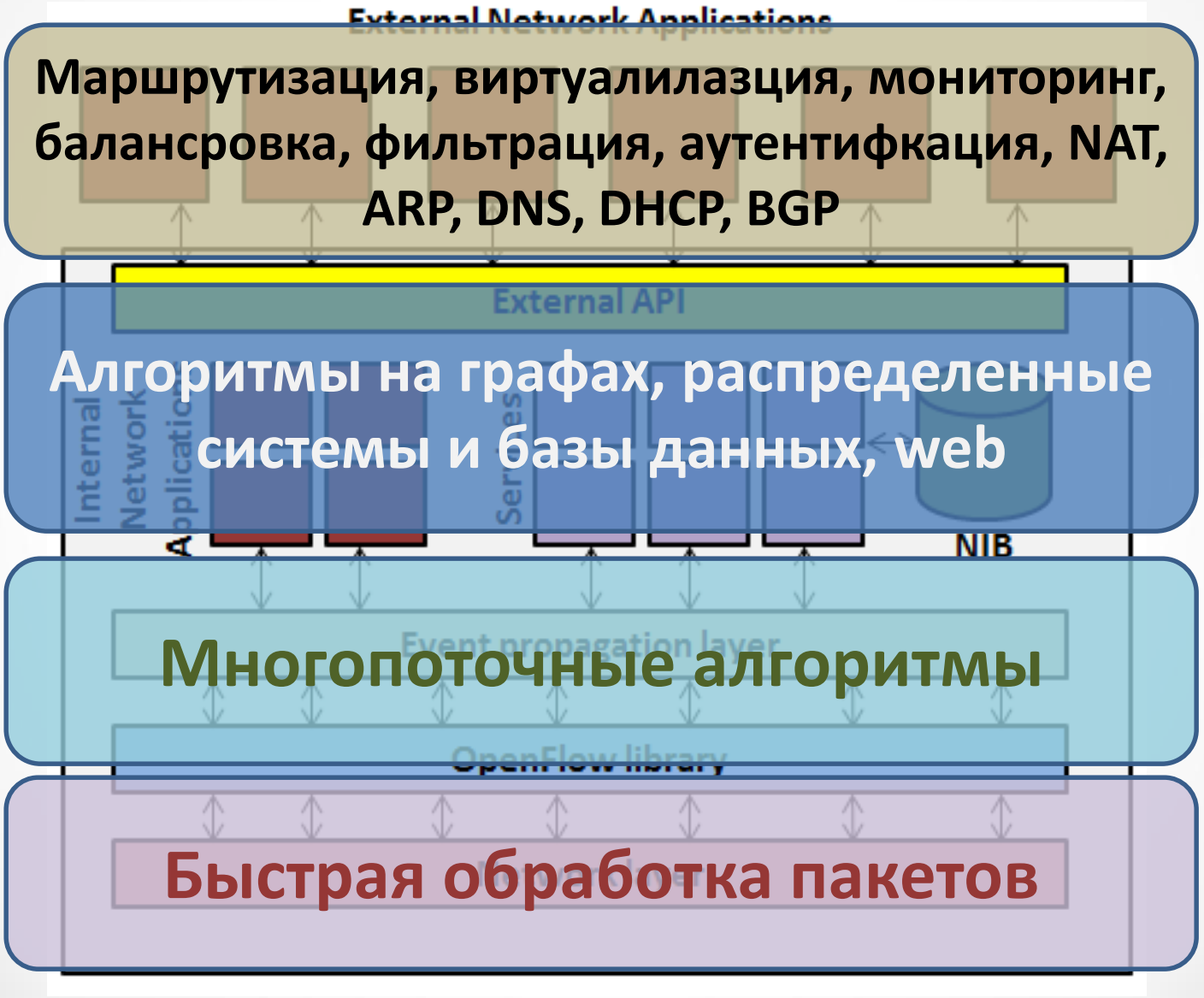


Требования к контроллеру ПКС

- Производительность
  - Пропускная способность
    - events per second
  - Задержка
    - us
- Надежность и безопасность
  - 24/7
- Программируемость
  - Функциональность: приложения и сервисы
  - Интерфейс программирования

- ЦОД требует обработку >10М событий в секунду
- Реактивные контроллеры более “чувствительные”

Архитектура OpenFlow контроллера



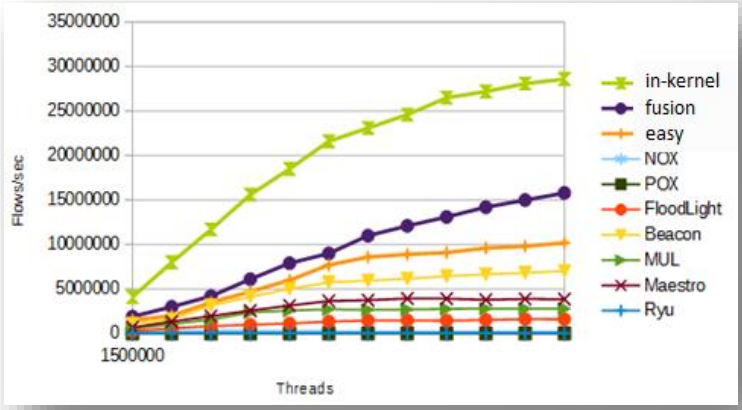
Часть III: Runos OpenFlow контроллер

Сетевая операционная система Runos

Система управления сетью первый российский SDN-контроллер RUNOS

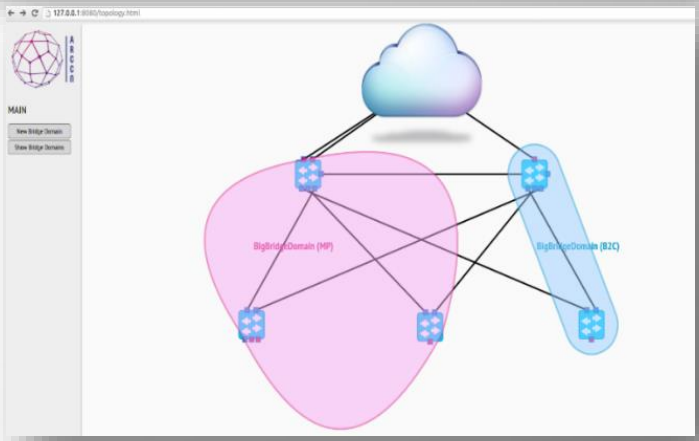
RUssian Network Operation System

Есть разные варианты контроллера с единой базой и различным набором сервисов и приложений



- **Открытая версия на Github** http://arccn.github.io/runos/
  - Своя база на C++11/14, а не Java
  - цель: упростить разработку сетевых приложений и не забывать о производительности
  - приложения: топология, маршрут, перестроение в случае обрыва, REST, WebUI, проактивная загрузка правил, резервирование Active-Passive

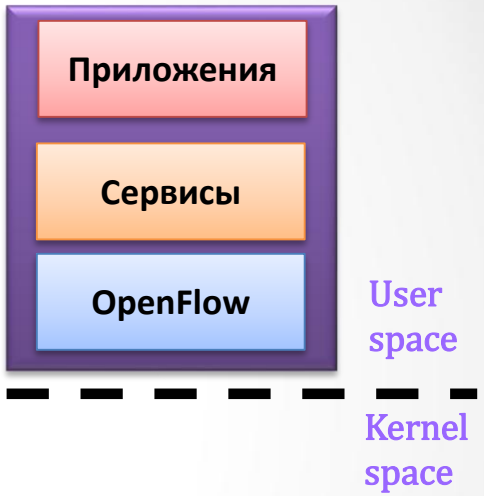
- **Внутренняя ядерная версия**
  - Супер-производительность 30 млн событий в секунду
  - Разработка приложений под заказчика



- **Внутренняя версия с приложениями под оператора связи**
  - База такая же, как и на Github. Заказчики сами могут разрабатывать приложения. Учиться по доступным материалам
  - Сервисы B2C, B2B (p2p, mp2mp, multicast, и т.п.)
  - Active-Standby режим

RUNOS: особенности

- **Алгоритмические политики (генерация правил)**
  - Распределение приоритетов, комбинация правил
  - LOAD, MATCH, READ абстракции
  - На основе MAPLE
- **Дружественный API на основе EDSL грамматики (низкоуровневые детали скрыты в системе выполнения – overloading, templates)**
  - “pkt[eth src] == eth addr”
  - “if (ethsrc == A || ethdst == B) doA else doB”
  - “test((eth_src & “ff0…..0”) == “….”)”
  - *“modify(ip_dst >> “10.0.0.1”)”*
  - decision are “unicast()”, “broadcast()”, “drop()”
- **Композиция приложений (параллельная и последовательная композация)**
  - dpi + (lb >> forwarding)



Features:

- Algorithmic policies (rule generation)
- Client-friendly API using EDSL grammar (low level details are hidden inside the runtime – overloading, templates)
- Modules composition (parallel and sequential composition)

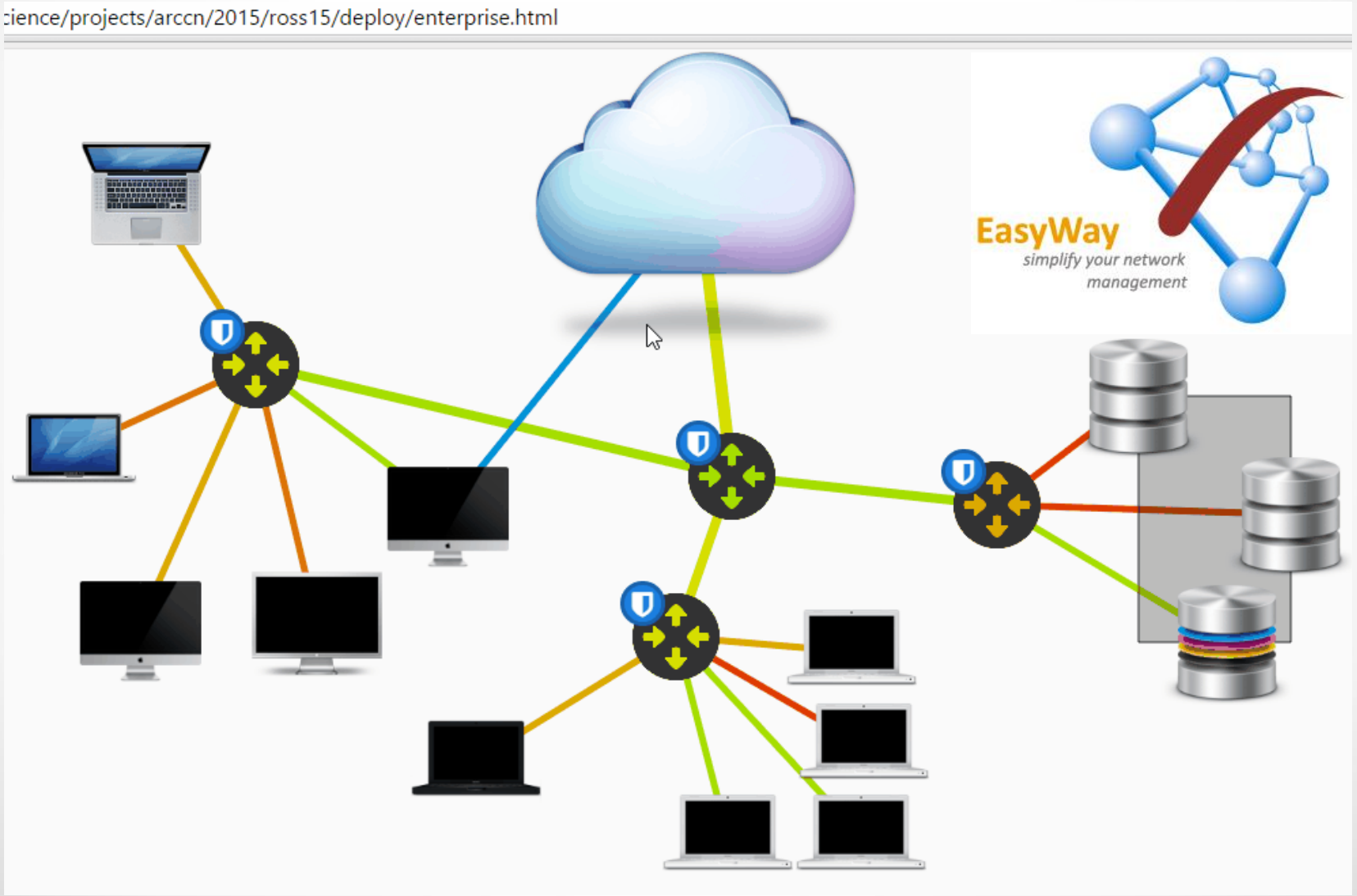
Реализация

Ключевые слова: C++11/14, QT, Boost (asio, proto, graph)

Основные сторонние компоненты:

- **libfluid project** (_base, _msg)
  - для взаимодействия со свитчами и разбор OpenFlow 1.3 сообщений
- **libtins**
  - разбор пакетов внутри OpenFlow сообщений
- **glog** (google log)
  - логирование, многопоточное
- **tcmalloc** (google performance tools)
  - альтернативная более быстрая реализация malloc/free
- **json11**
  - разбор конфигурационного файла
- **boost graph**
  - Хранение топологии, поиск маршрута

Пример графического интерфейса EasyWay



Описания релизов

Сейчас версия **0.5**

- ядро контроллера
- построение топологии
- построение маршрута через всю сеть
- первая версия системы генерации правил
- Rest API (совместимый с Floodlight)
- WebUI (мониторинг загрузки, просмотр таблиц, удаление и добавление правил)
- Проактивная загрузка правил
- Холодное резервирование
- ARP кеширование

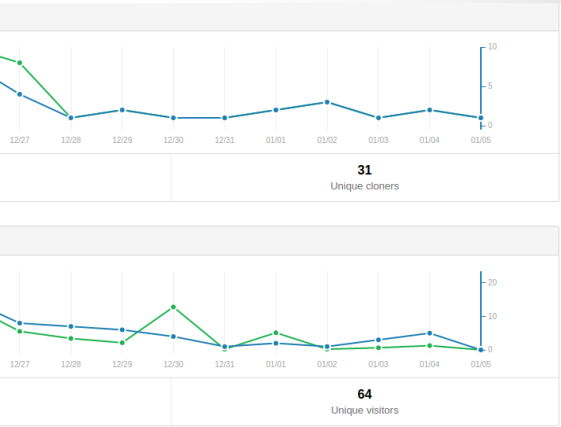
Описания релизов

Версия **0.6** - следующий большой релиз (апрель)

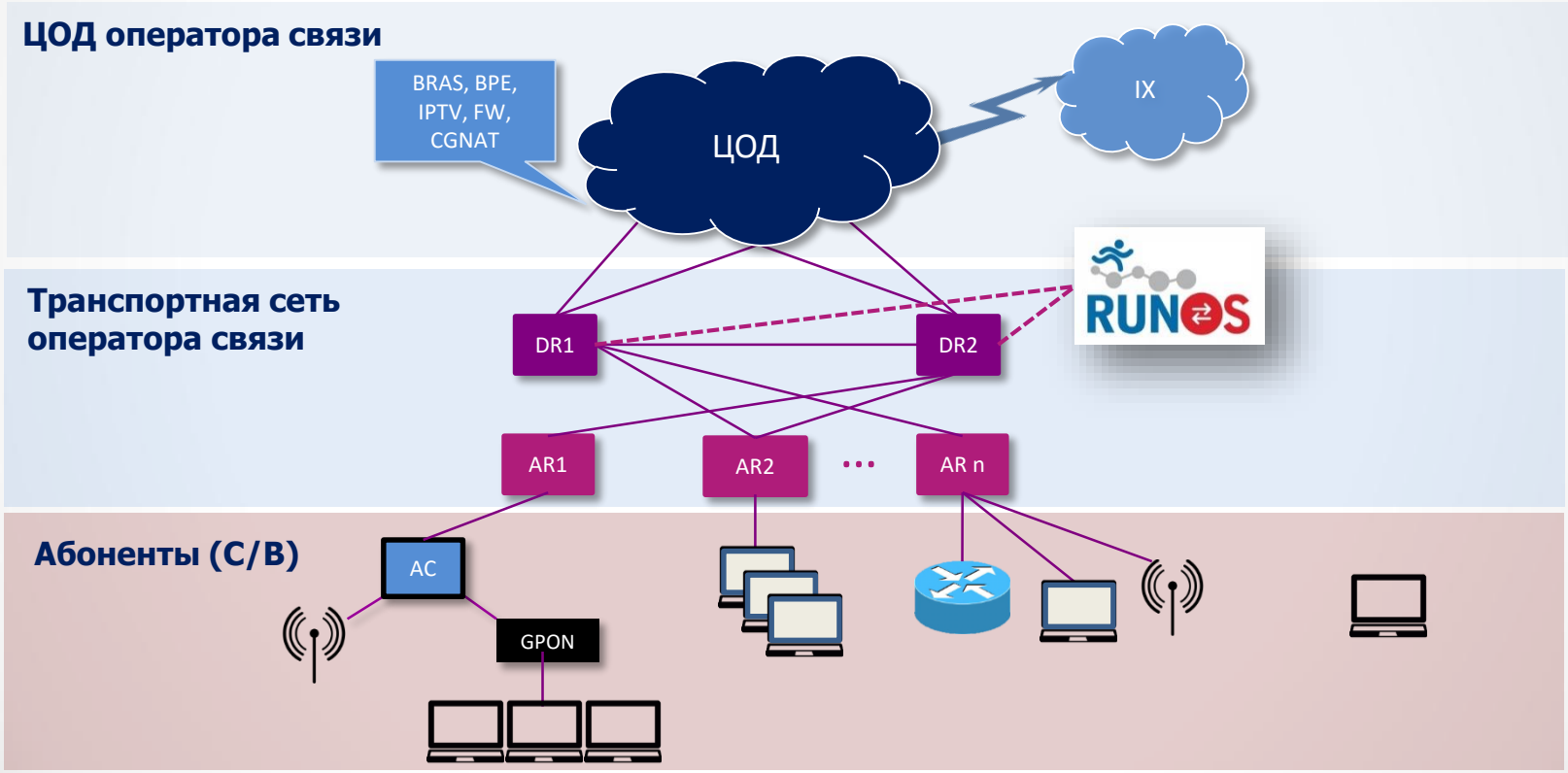
- **Полное обновление структуры ядра контроллера**. Нет привязка к конкретной версии протокола OpenFlow. Своя модель, расширяемая под любые новые поля, в том числе и специфические для оборудования.
- **Пакетная грамматика для сетевых приложений**. Упрощает разработку новых приложений.
- **Обновление системы генерации правил —** повышена скорость работы и улучшена генерация правил (по количетсву правил и числу приоритетов).
- Возможность статического связывания модулей.
- Система тестов.

Проект с открытым исходным кодом

- Исходный код http://arccn.github.io/runos/
  - Apache, version 2.0
- Tutorial (Readme.md)
  - Как установить, запустить, написать свое первое приложение
- Виртуальная машина
  - Уже собранный контроллер
  - Средства для работы с OpenFlow
- Список рассылки
  - Google group **runos-ofc**



Общая схема сети оператора связи



Создание сервисов сети операторов

Сервисы:

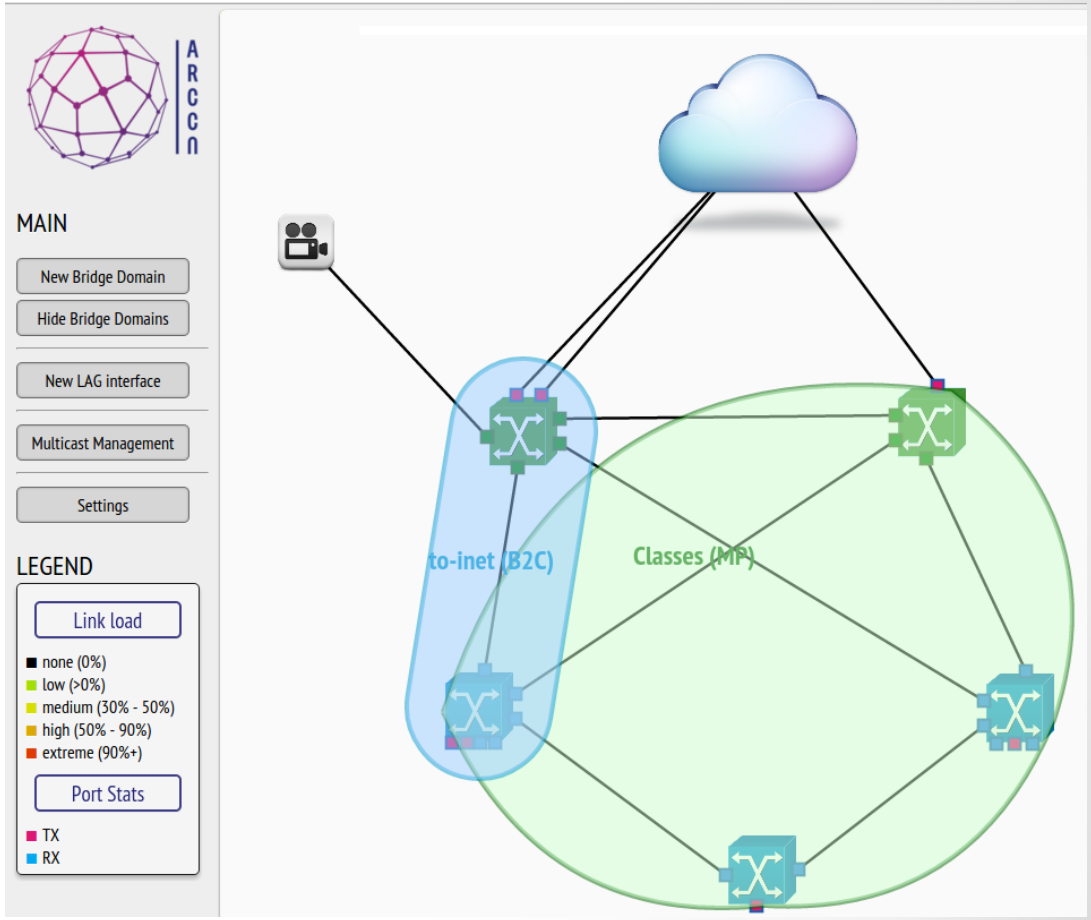
1. B2C, P2P, MP
2. Multicast
3. Storm Control
4. LAG/LACP
5. InBand

Резервирование сервисов:

1. Active-Standby Controller
2. Fast failover резервирование
3. Активное резервирование

Качество доступа (QoS):

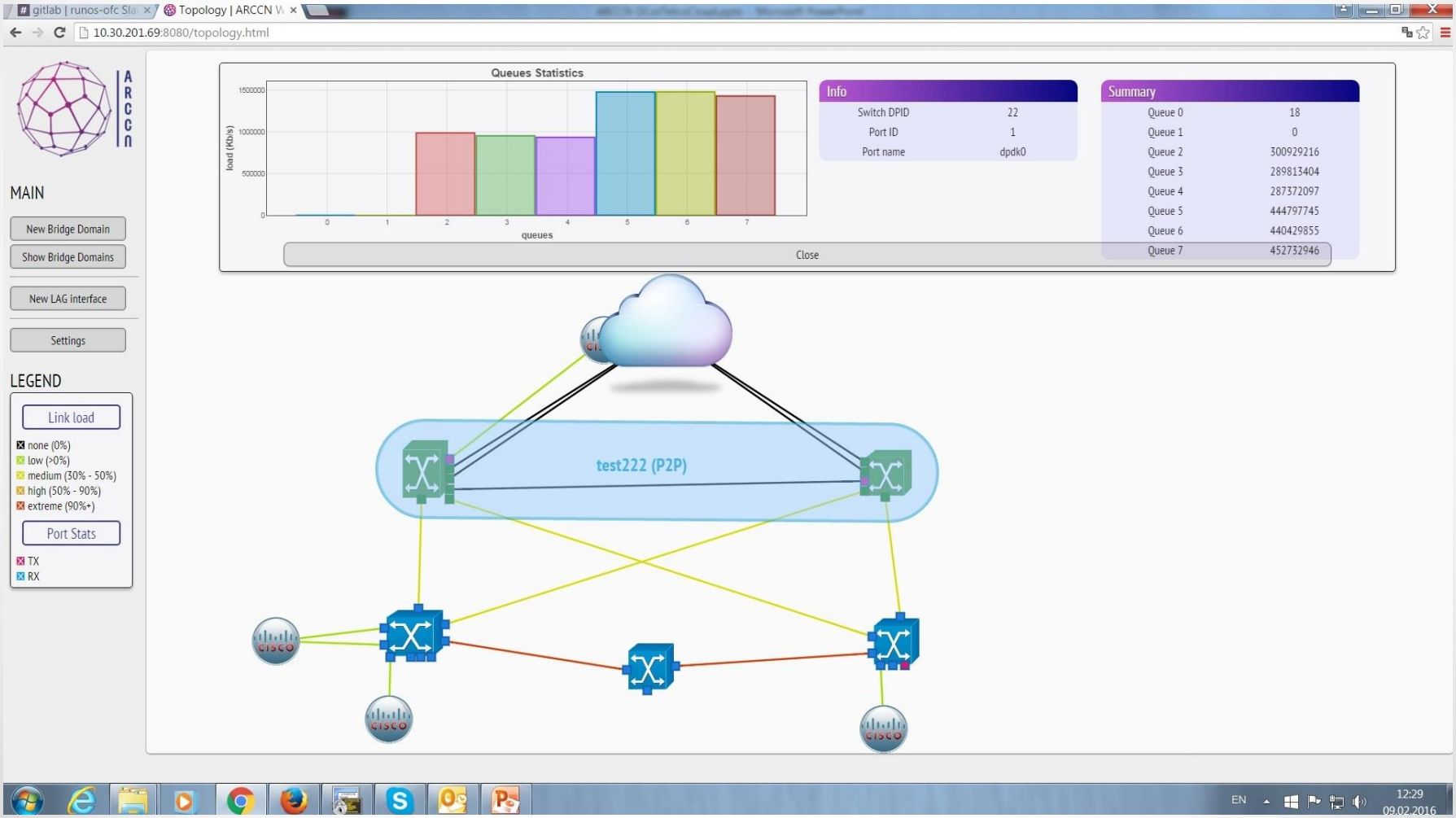
1. Priority Queuing, WRR
2. Rate-Policy, Ingress QoS, metering
3. очереди на интерфейсах



QoS и мониторинг трафика

Система управления сетью первый российский SDN-контроллер RUNOS

И набор приложений для управлению сетью



Open Source

- Два типа OpenSource проектов:
  - ради Идеи: “Free as in Freedom”
  - продаем свои компетенции, а не продукт
    - доработка под нужды заказчика,
    - продавать advanced версии и плагины (eg, приложения для Runos)
    - community вокруг (семинары, обучение)
- Важна лицензия (*): Apache, BSD или GPL, Eclipse, проприетарные лицензии, перелицензирование за деньги
- Угрозы:
  - Скопируют и будут продавать под другим названием (eg, runos-ng)
  - Будут дорабатывать своими силами (для компаний с большим R&D)

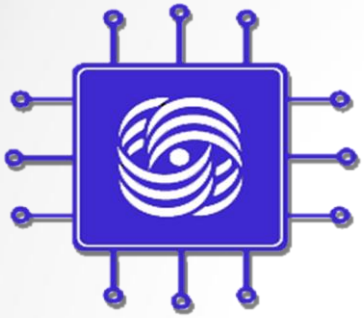
* http://www.slideshare.net/gerasiov/license-44646637

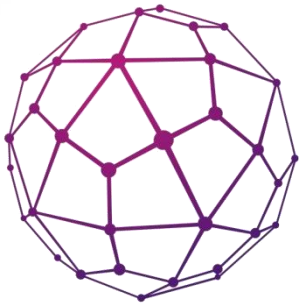
Заключение

- 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





ЦЕНТР
ПРИКЛАДНЫХ
ИССЛЕДОВАНИЙ
КОМПЬЮТЕРНЫХ
СЕТЕЙ

Спасибо за внимание!