

Актуальность. Сценарии 5G/IMT-2020

Усовершенствованная подвижная широкополосная связь

(Enhanced Mobile Broadband) EMB

Гигабайт в секунду



Трехмерное видео, экраны UHD

"Умный" дом/здание



Работа и развлечения в облачных сервисах

Дополненная реальность

Голос



Автоматизация производства

"Умный" город



5G/IMT-2020



Критически важные приложения



Беспилотный автомобиль

Massive Machine Type Communications (MTC)



Ultra-reliable and Low Latency Communications (URLLC)

Сверхнадежная связь с малой задержкой

Крупномасштабные системы межмашинной связи

Ресурс: ITU-R M.2083-0: «Концепция IMT – Основы и общие задачи будущего

развития IMT на период до 2020 года и далее»

Актуальность. Требования к сетям 5G/IMT-2020

Большие данные и Машинное обучение

Новые методы мониторинга и управления трафиком
(управление на основе знаний о всех потоках и прогнозной аналитики)

	Latency	Throughput	Connections	Mobility	Network Architecture
5G	1 ms E2E Latency	10G bps Per Connection	1,000K Connections Per km ²	500 km/h High-speed Railway	LINP Ability Required
GAP	30~50x	100x	100x	1.5x	NFV/SDN
LTE	30~50ms	100Mbps	10K	350Km/h	Inflexible

Ваш логотип

Ресурс: NGMN 5G White Paper

▶ **ITU** → → **IMT/2020**

▶ **3GPP** → → **5G**

функции и требования служб 5G, определенных в версии 14 и 15 3GPP, и технические отчеты МСЭ-Р для IMT/2020.

5G

- ▶ При массовом увеличении разнообразных беспроводных устройств (например, смартфонов, устройств МТС и носимых устройств) подключение и объем трафика становятся проблемой создания для сотовой системы 5G.
- ▶ Ожидается, что трафик данных в 2020 году будет в 200 раз больше, чем в 2010 году, а на 2030 год поток данных будет в 20,000 раз больше, чем на 2010.

5G – Необходимые условия

▶ Одной из основных целей создания сотовой системы 5G является достижение скорости передачи данных до 10 Гбит / с.

▶ Кроме того, для реализации сотовой системы 5G требуется достигать следующих параметров:

- высокая пропускная способность,
- сверхнизкое время ожидания,
 - высокая надежность,
- высокая возможность соединения и
 - высокая подвижность.

Энергосберегающие технологии

- ▶ Для преодоления проблемы загрузки трафика и проблемы подключения; и для достижения ожидаемой скорости передачи данных со сверхнизкой задержкой, сотовая система 5G должна использовать новые технологии в различных местах сети.
- ▶ В базовой сети должны использоваться некоторые технологии, такие, как **программно-конфигурируемые сети (SDN)** и **виртуализация сетевых функций (NFV)**.
- ▶ Другие технологии используются на границе сети, чтобы разгрузить базовую сеть и также обеспечить низкое время ожидания. Эти технологии похожи на MEC и коммуникацию D2D.

Категории вариантов использования 5G

- 1- Сверхширокополосная мобильная связь (eMBB);
- 2- Массовая межмашинная связь (mMTC); а также
- 3- Сверхнадежная межмашинная связь (uRLLC).



Сценарий развертывания

1. Внутренняя точка доступа;
2. Плотность города;
3. Покрытие сел;
4. Городское макро-покрытие;
5. Высокая скорость;
6. Городское покрытие для массивных соединений;

Сценарий развертывания

7. Экстремальное дальнейшее покрытие в районах с низкой плотностью;
8. Магистраль;
9. Городская сетка;
10. Коммерческий воздух как основа;
11. Легкий летательный аппарат; и
12. Спутниковое расширение для наземного.

Интернет вещей

- Концепция Интернета вещей
- Новая модель взаимодействия

2. Сервисы IoT

- Новые типы сервисов
- Проблемы реализации сервисов
Интернета вещей
- Примеры практической реализации
коммерческих продуктов в сфере
IoT



Интернет вещей. Концепция. Определения

ITU-T, Y.2060

Интернет вещей — глобальная инфраструктура для информационного общества, которая обеспечивает возможность предоставления более сложных слуг путем соединения друг с другом (физических и виртуальных) вещей на основе существующих совместимых информационно-коммуникационных технологий.

Вещь - физический, так и виртуальный объект, при этом являющийся предметом, который может быть идентифицирован и интегрирован в информационно-коммуникационную инфраструктуру.

Рекомендации ИТУ:

- Y.2061
- Y.2062
- Y.2063



Интернет вещей. Концепция. USN

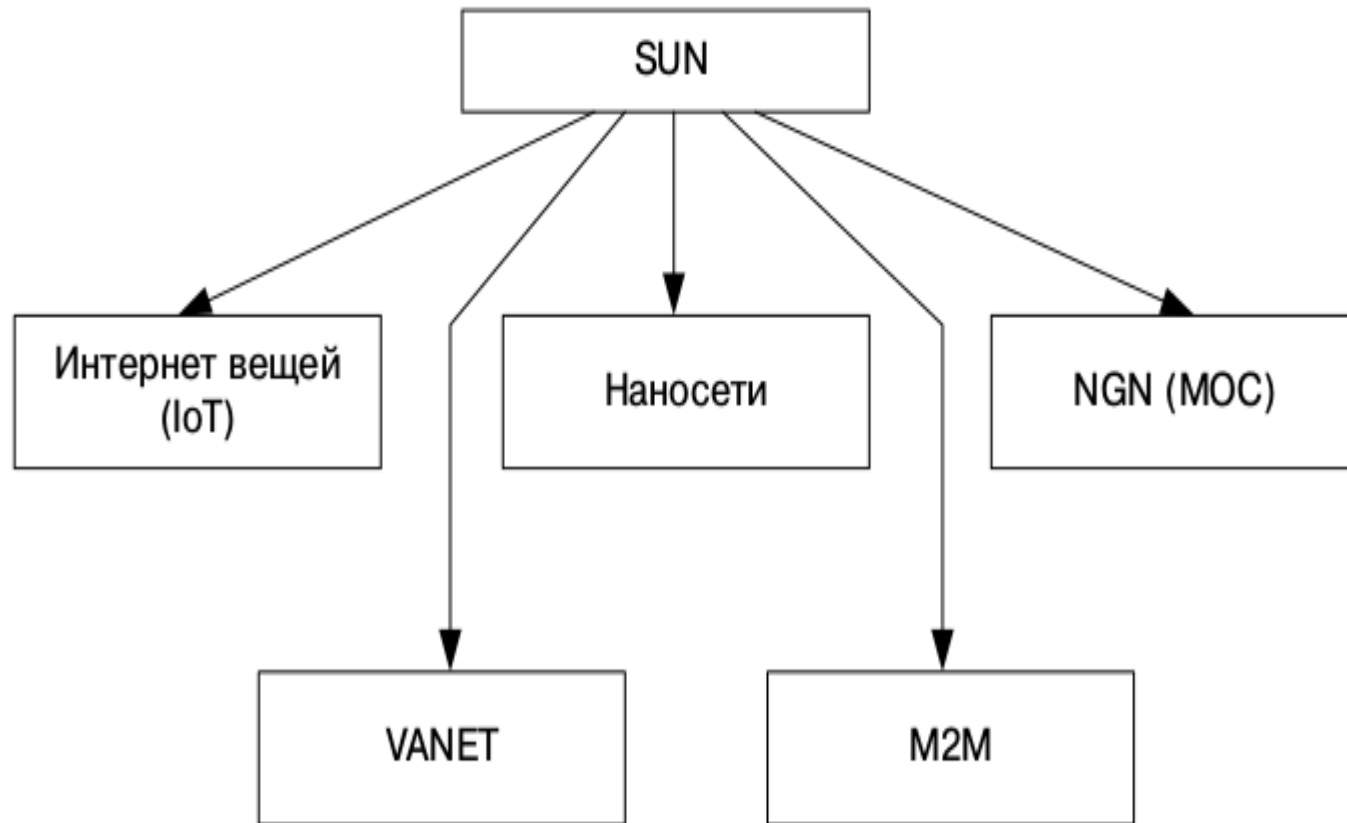
USN — Ubiquitous Sensor Network -
(всепроницающие сенсорные сети)



Ваш логотип

Интернет вещей.

SUN — Smart Ubiquitous Networks - всепроникающие сети



Интернет вещей. Новая модель взаимодействия

Контент: ТВ, YouTube, сайты, соцсети.....



Цель — поставка подходящего в правильном контексте на любое



Ваш логотип

Интернет вещей. Поиск в эпоху IoT?

- Результаты поиска с помощью смарт-устройств будет представлен на ма
- Поисковые запросы станут более простыми и естественными;
- Результаты поиска определяются:
 - Место положения пользователя
 - Пути перемещения
 - Финансовое положение

грубо говоря — результат одного и того же запроса в кухне и спальне (

- Поиск будет персонализирован
- Релевантность будет определяться в контексте ваших предпочтений, пр
- Поиск станет проактивным (научатся предоставлять результаты до того, как вы успеете ввести запрос)



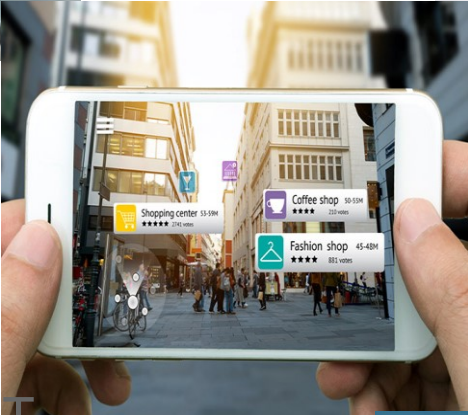
Требования
к качеству трафика

Сервисы IoT. Новые типы сервисов

Тактильный интернет



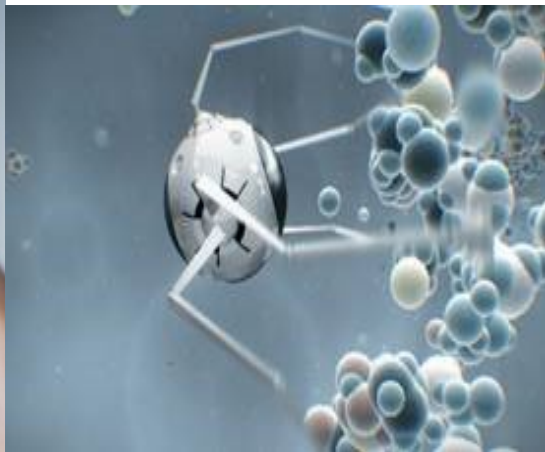
AR (дополненная реальность)



VR (виртуальная реальность)

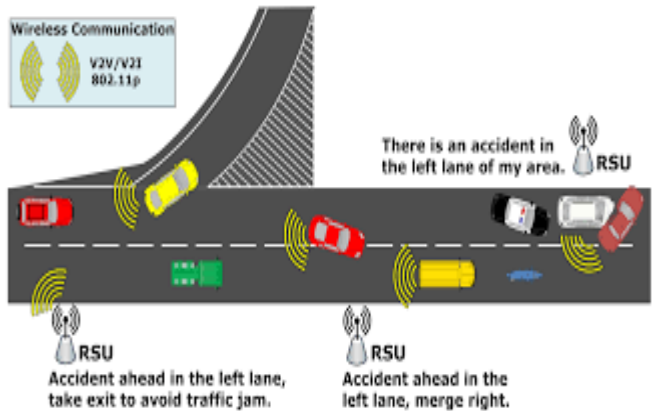


Медицинские сети



Индустрия 4.0

VANET



Smart Cities



Ваш логотип

Примеры из практики

Города

- Сигнапур («умная дорожная система»)
- Хельсинки («программа повышения доступности информ. ресурсов»)
- Санкт-Петербург / Москва (IoT в ЖКХ, умные системы видеонаблюдения, управление светофорами)

• и др.

Облачные системы менеджмента

- Microsoft Azure
- Go+
- io.Thinger
- Intel Dashboard
-



Yandex



Google

«Умные» сервисы:

- Приложения и сервисы Y
- Приложения и сервисы C
- Uber
-



INTERNET
OF THINGS

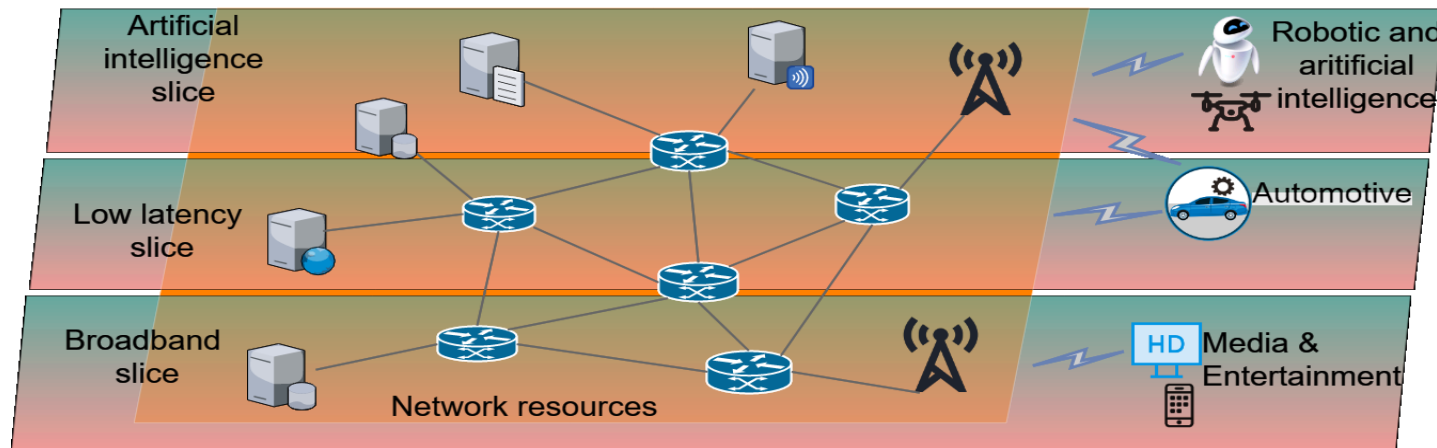
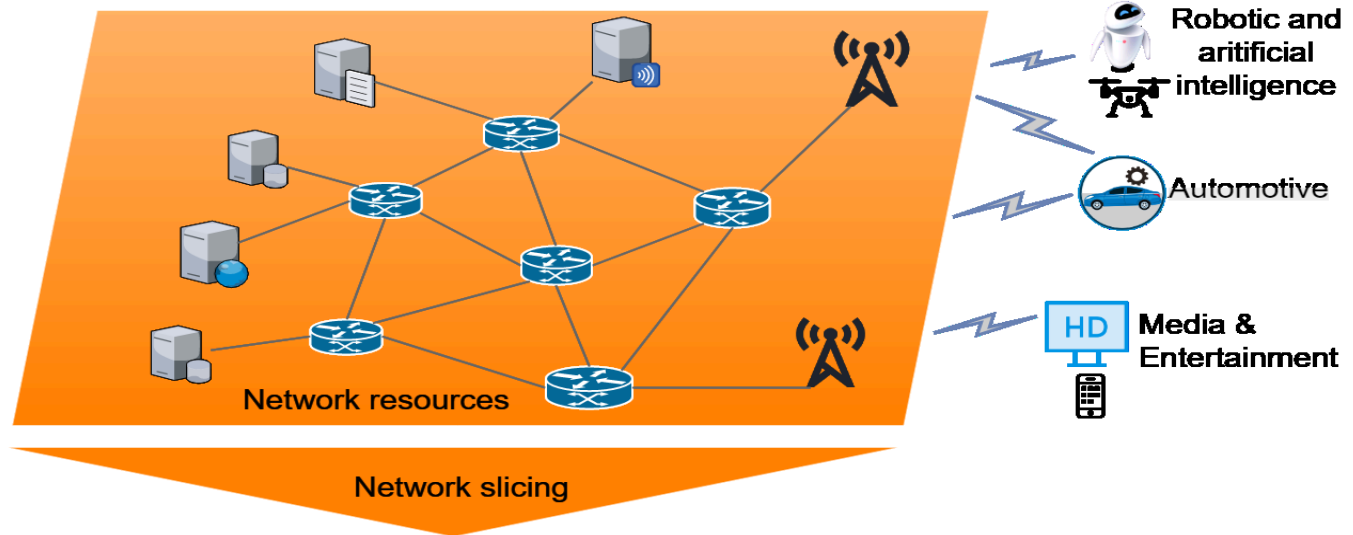


Ваш логотип

А. Сетевой слайсинг

- Сетевой слайсинг - это вариант использования 5G, который эффективно обеспечивает работу нескольких сетей на общей платформе.
- Используя сетевой слайсинг, сотовые операторы могут создавать и управлять независимыми приложениями через общую единую физическую сетевую инфраструктуру
- На основе проекта 3GPP и NGMN сетевой слайсинг- это механизм, который позволяет операторам сетей 5G предоставлять широкий спектр различных услуг для различных секторов рынка. Эти службы могут выполнять различные и неоднородные требования с точки зрения надежности, охвата, емкости и задержки.

А. Сетевой слайсинг



В. Тактильный интернет

- ▶ Тактильный интернет создает интерактивную систему реального времени между человеком и машиной и вносит новую эволюцию в сообщение между человеком и машиной (H2M).
- ▶ Тактильный Интернет позволит удаленно передавать физические привычки и, таким образом, вводит новый переход парадигмы к сетям, основанным на навыках, вместо сетей на основе контента.

Тактильный интернет

Master Domain



Network Domain



Slave Domain



С. Биосвязь

- ▶ Биосвязь - это вариант использования 5G, который рассматривает медицинские службы телеметрии через систему 5G.
- ▶ Носимые датчики будут использоваться, чтобы измерить медицинские показатели, такие, как кровяное давление человека, ритм сердца, температура и глюкоза в крови, и служба биосвязи должна упростить непрерывную и автоматическую телеметрию этих медицинских показателей.

С. Биосвязь

► Существует множество требований для предоставления услуг биосвязи, которые включают следующее:

1. Очень высокая надежность;
2. Очень высокая безопасность;
3. Высокие возможности подключения;
4. Скорость передачи данных «от низкой к высокой»: для услуг, поддерживающих видео, может потребоваться высокая скорость передачи данных; и
5. Требования к пользовательскому оборудованию (например, низкая мощность и низкая сложность).

D. Спасательная связь/ Природные катастрофы

- ▶ Этот случай использования 5G в основном касается сообщений в ситуациях катастроф и стихийных бедствий. Системы 5G должны обеспечивать базовую и надежную связь в таких ситуациях, как наводнения, ураганы, цунами и землетрясения.
- ▶ В катастрофических ситуациях должен быть обеспечен минимальный уровень коммуникационных услуг, включающий себя передачи речи и отправку SMS.
- ▶ Основным требованием этих услуг является энергоэффективность, так как пользовательское оборудование должно потреблять энергию на минимальном уровне, чтобы поддерживать мощность устройства в течение более длительного времени.

Е. Подключенные транспортные средства

► Это вариант использования услуг, связанных с развертыванием автономных транспортных средств, и тем, как между соседними транспортными средствами должен происходить обмен сообщениями.

► Развертывание автономных транспортных средств потребует полного охвата всех географических регионов. Развитие услуг связанных транспортных средств будет иметь множество требований, которые включают в себя следующее:

1. Ультранизкая сквозная задержка 1 мс;
2. Очень высокая надежность, близко к 100%;
3. Полный охват;
4. Очень высокая системная доступность, близко к 100%;
5. Поддержка многоадресной передачи и широковещательных передач;
6. Очень высокая скорость передачи данных, конечно для видео сценариев;
7. Очень высокая точность позиционирования;
8. Высокая гибкость, это происходит из-за изменения топологии; и
9. Очень высокая мобильность.

Г. Автоматизация производства

- ▶ Включает в себя автоматизацию производства и автоматизацию процессов.
- ▶ Применения для автоматизации производства требуют контроля за замкнутым циклом, однако для автоматизации процесса требуется только система связи с открытым контуром для мониторинга и отслеживания всех производственных процессов.
- ▶ Службы автоматизации будут обрабатываться главным образом за счет производства роботов и автоматических машин.

Е. Автоматизация производства

▶ Услуги автоматизации процесса будут обрабатываться большим количеством датчиков порядка 10 тысяч, которые будут распределяться по этапам производства и процессам.

▶ Требования к обслуживанию приложений автоматизации промышленных предприятий включают следующее:

1. Высокая надежность;
2. Поддержка полезной нагрузки между 50 - 100 байтами за транзакцию;
3. Уровень очень высокой степени безопасности;
4. Крайняя низкая задержка; и
5. Ограничения связи в коротком диапазоне.

Приложения дополненной реальности



Ваш логотип

ОСНОВНЫЕ ПОКАЗАТЕЛИ РЕЗУЛЬТАТИВНОСТИ РАБОТЫ (КРІ)

► Различные сквозные КРІ для 5G и IMT / 2020 определены в версиях 14 и 15 3GPP и технических отчетах МСЭ-R.

► ITU определяет различные условия, связанные с качеством обслуживания системы 5G в ITU-T в рекомендации E.800.

Основные категории КРІ для системы 5G включают в себя:

- 1.общие КРІ;
- 2.доступность КРІ;
- 3.целостность КРІ; и
- 4.использование КРІ.

ОСНОВНЫЕ ПОКАЗАТЕЛИ РЕЗУЛЬТАТИВНОСТИ РАБОТЫ (КРІ)

1. Общий КРІ:

- Пиковая скорость передачи данных.
- Пиковая спектральная эффективность.
- Пропускная способность.

2. Доступность КРІ:

- Успешность регистрации одного экземпляра слоя сети.
- Зарегистрированные подписчики сетевого и сетевого слоев через AMF.
- Зарегистрированные абоненты сетевого и сетевого слоев через UDM.

ОСНОВНЫЕ ПОКАЗАТЕЛИ РЕЗУЛЬТАТИВНОСТИ РАБОТЫ (КРІ)

3. Целостность КРІ:

- Полная закрытость сети 5G.
- Пропускная способность RAN UE.
- Пропускная способность по потоку в интерфейсе N3.
- Пропускная способность по потоку в интерфейсе N3.
- Пропускная способность по потоку для одного сетевого слоя.
- Пропускная способность по потоку для сетевого и слайсового сетевого слоев

4. Использование КРІ:

- Использование виртуализованного ресурса экземпляра сетевого слоя.
- Среднее количество сеансов PDU для сетевого и слайсового сетевого слоев.

Мобильные граничные вычисления (MEC)

Просто MEC можно определить как способ перемещения облачных вычислительных возможностей на границу мобильных сетей. Перемещение облачных вычислений на границу мобильной сети дает много преимуществ, которые можно суммировать в следующих пунктах:

- 1- уменьшает время задержки при передаче данных,
- 2- обеспечивает эффективный способ разгрузки данных, передаваемых в основную сеть,
- 3- обеспечивает высокую пропускную способность и
- 4- вводит новые службы и приложения, получающие доступ к информации о сетевом контексте.

Мобильные граничные вычисления (МЕС)

Переход от больших, массивных и дорогостоящих центров обработки данных в небольшие распределенные облачные блоки на основе небольшой аппаратной платформы откроет путь для достижения требуемого ограничения времени для тактильной реализации.

Существует несколько расположений для серверов МЕС, таких как:

1- Серверы облаков подключены к базовой станции LTE (eNB).

2- Облачный модуль может быть помещен в контроллер радиосети 3G / 4G (RNC).

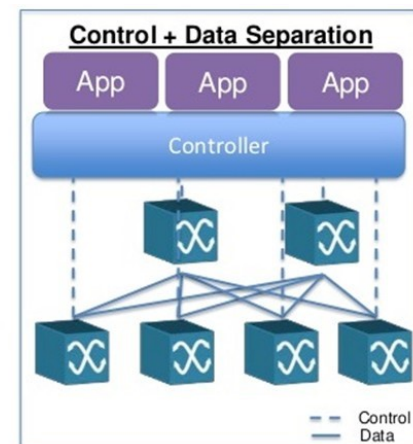
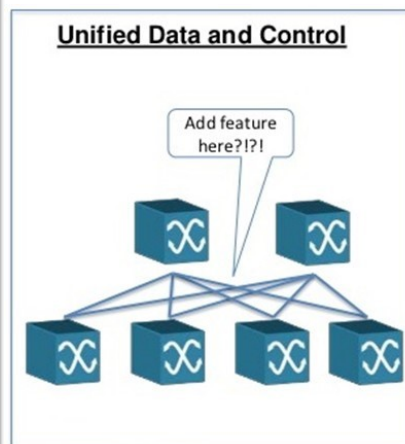
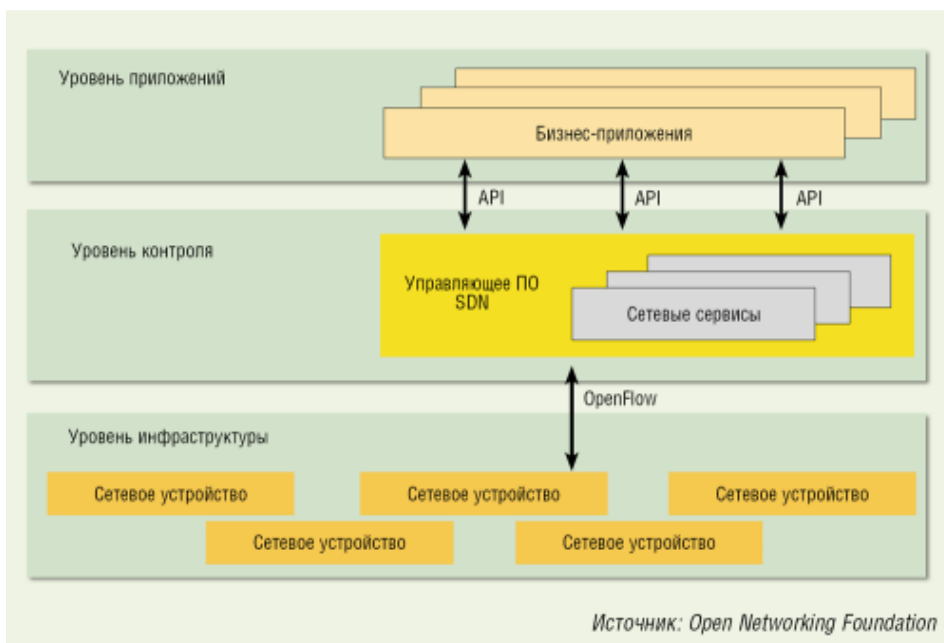
3- Модуль облака может быть подключен к нескольким сайтам (несколько eNB).

4- Облачный модуль может находиться на границе базовой сети.

Новый подход - SDN

Программно-конфигурируемая сеть (SDN) — новая концепция, предполагающая разделение функций управления и передачи трафика, что позволит изменить методы построения сетей и их эксплуатации.

Фонд открытых сетевых технологий (Open Networking Foundation, ONF) : SDN — динамичная, управляемая и адаптируемая сетевая архитектура, в которой разделены уровни управления сетью и передачи данных, что обеспечивает программное управление сетью и абстрагирование/изоляция (уровня) сетевой инфраструктуры от (уровня) приложений и сетевых услуг/сервисов .



Стандарты.

В 2011 г. компании Facebook, Deutsche Telekom, Microsoft, Verizon и Yahoo! организовали консорциум **ONF (Open Networking Foundation)** с целью развития технологий SDN в целом

В настоящее время ONF уже разработаны следующие документы:

спецификация протокола и коммутатора OpenFlow, текущая версия 1.4.0 (08/2013), предыдущие версии 1.0.x, 1.3.x и расширения к ней; версия 1.5.0 планировалась к выходу в январе 2014 г. , но пока отсутствует

спецификация протокола конфигурации коммутаторов OpenFlow и построения сетевой среды OF-Config, версия 1.2 (2014)

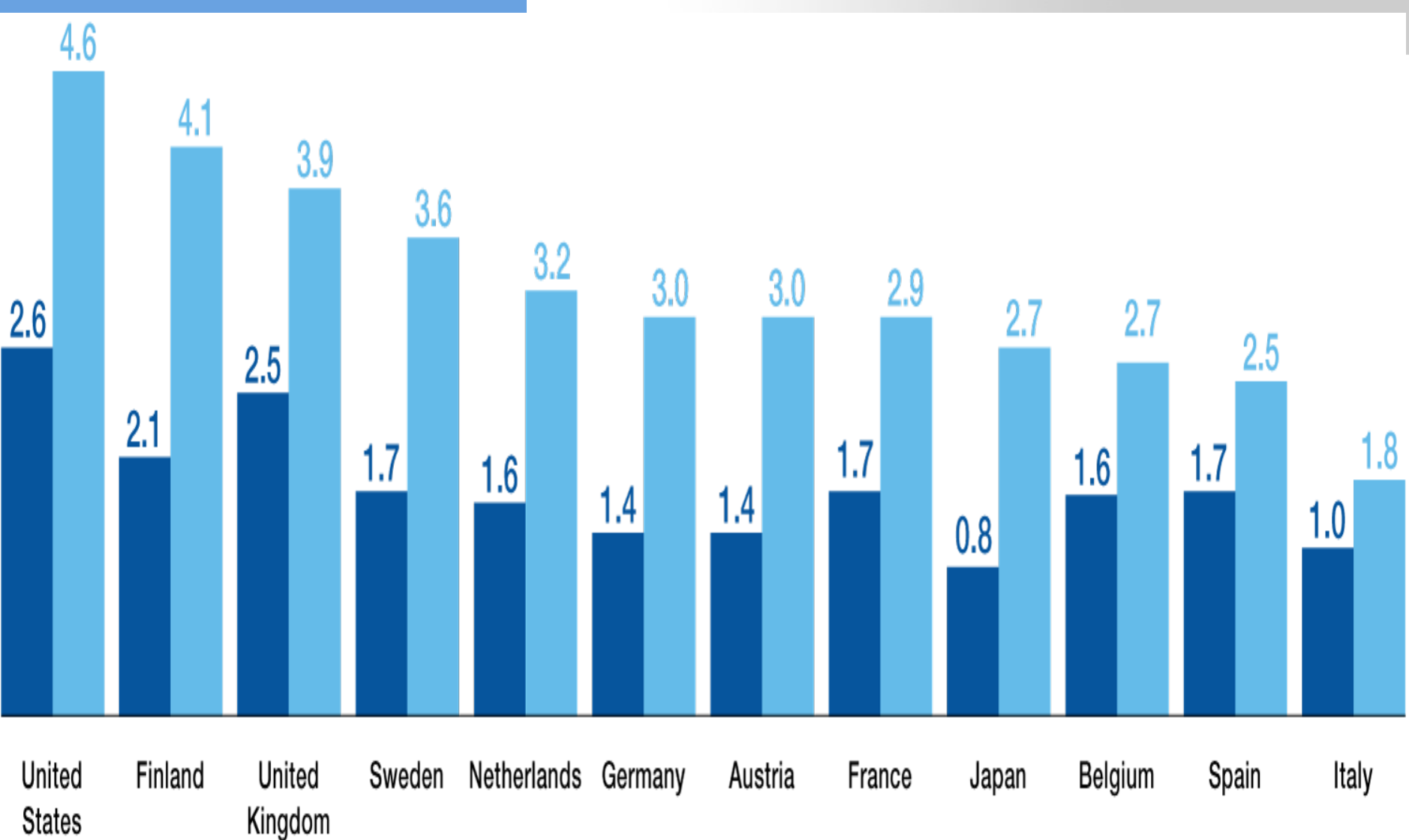
структура оповещений о событиях OpenFlow, версия 1.0

спецификация тестов на соответствие стандарту OpenFlow Switch, версия 1.0.1 . и протокола OpenFlow в частности.



OPEN NETWORKING
FOUNDATION

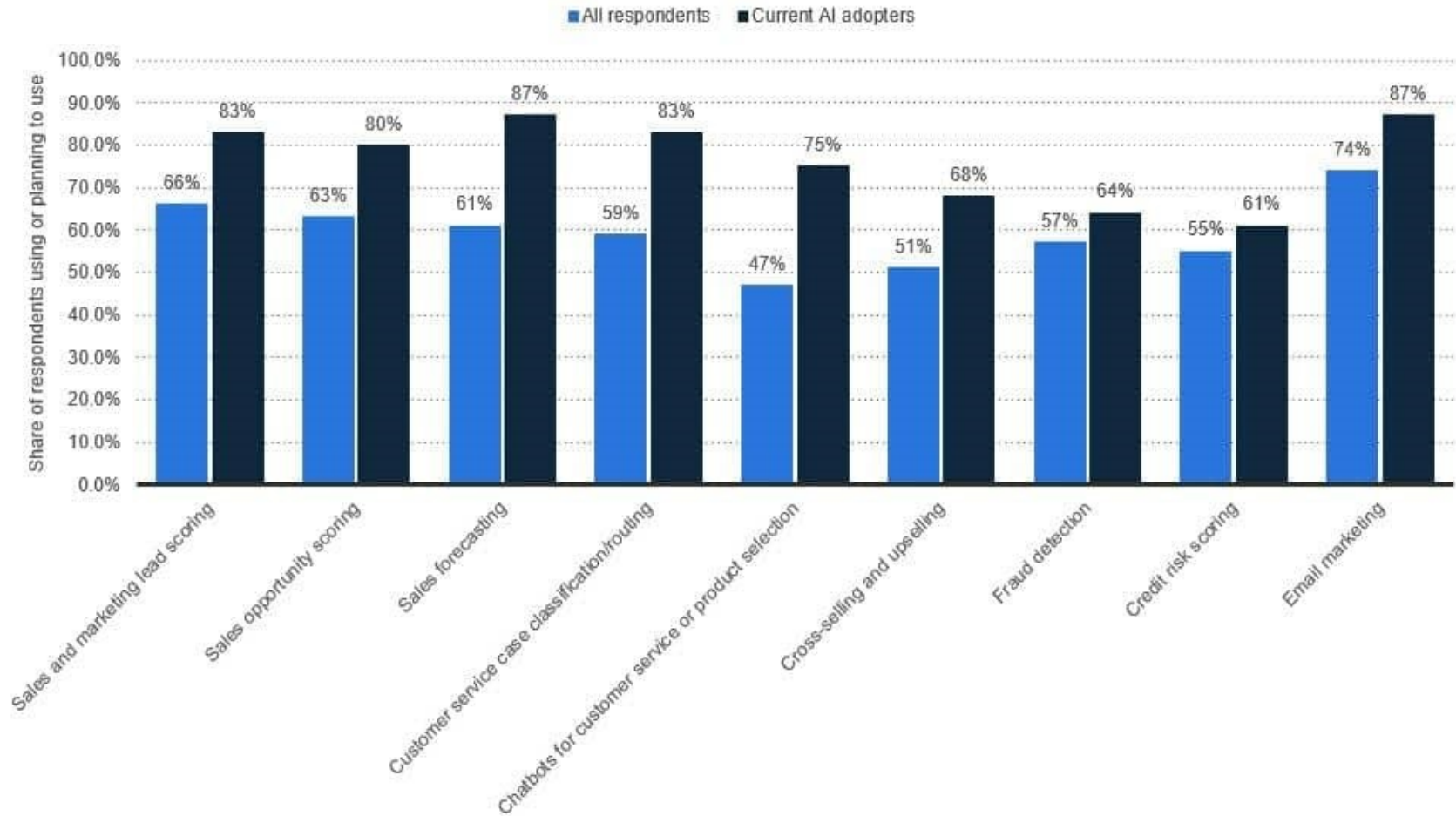
Искусственный Интеллект



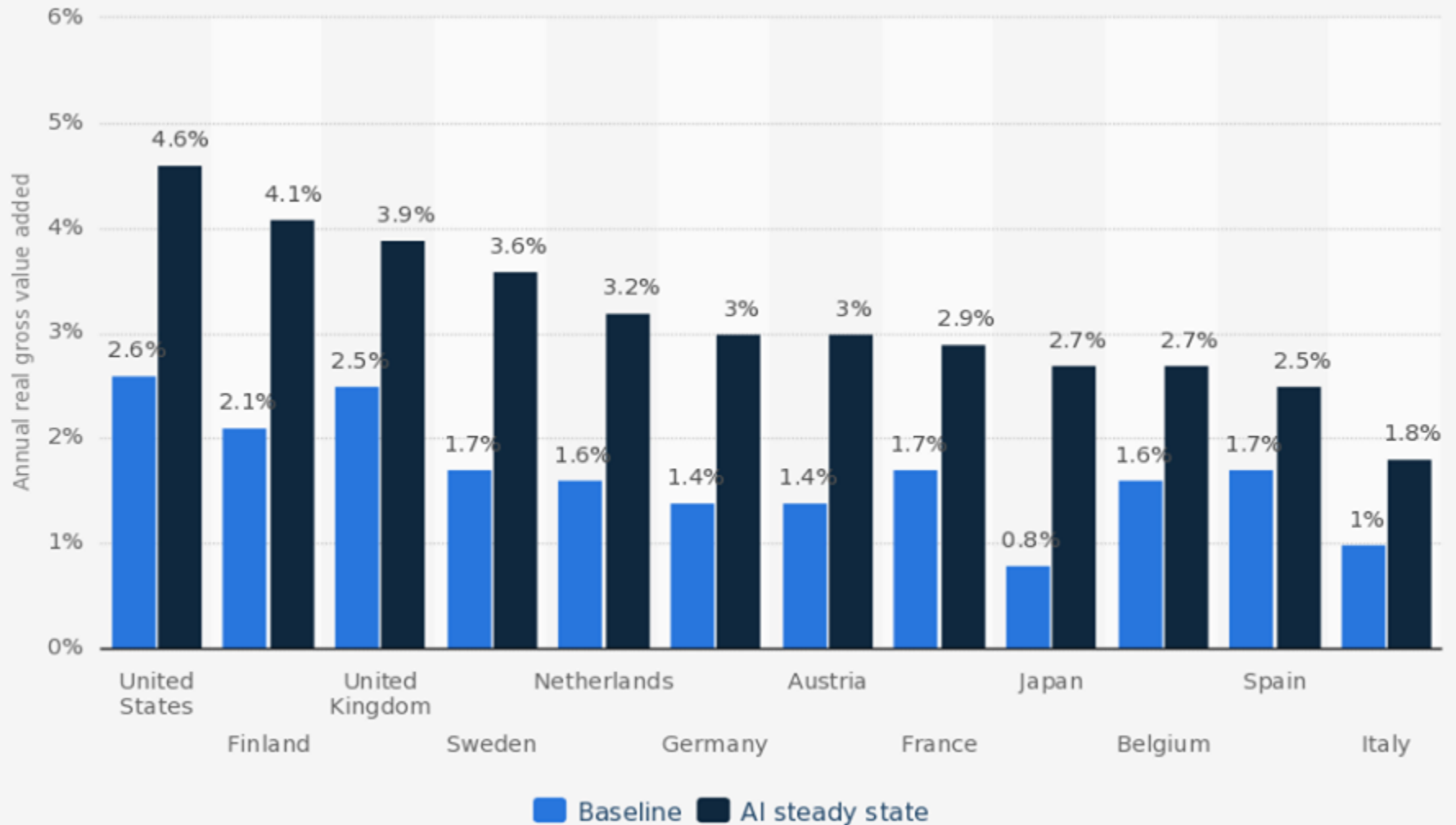
Baseline
 AI steady rate

Annual growth rates in 2035 of gross value added (a close approximation of GDP), comparing baseline growth in 2035 to an artificial intelligence scenario where AI has been absorbed into the economy.

Adoption of specific artificial intelligence (AI) use cases in 2017, by category



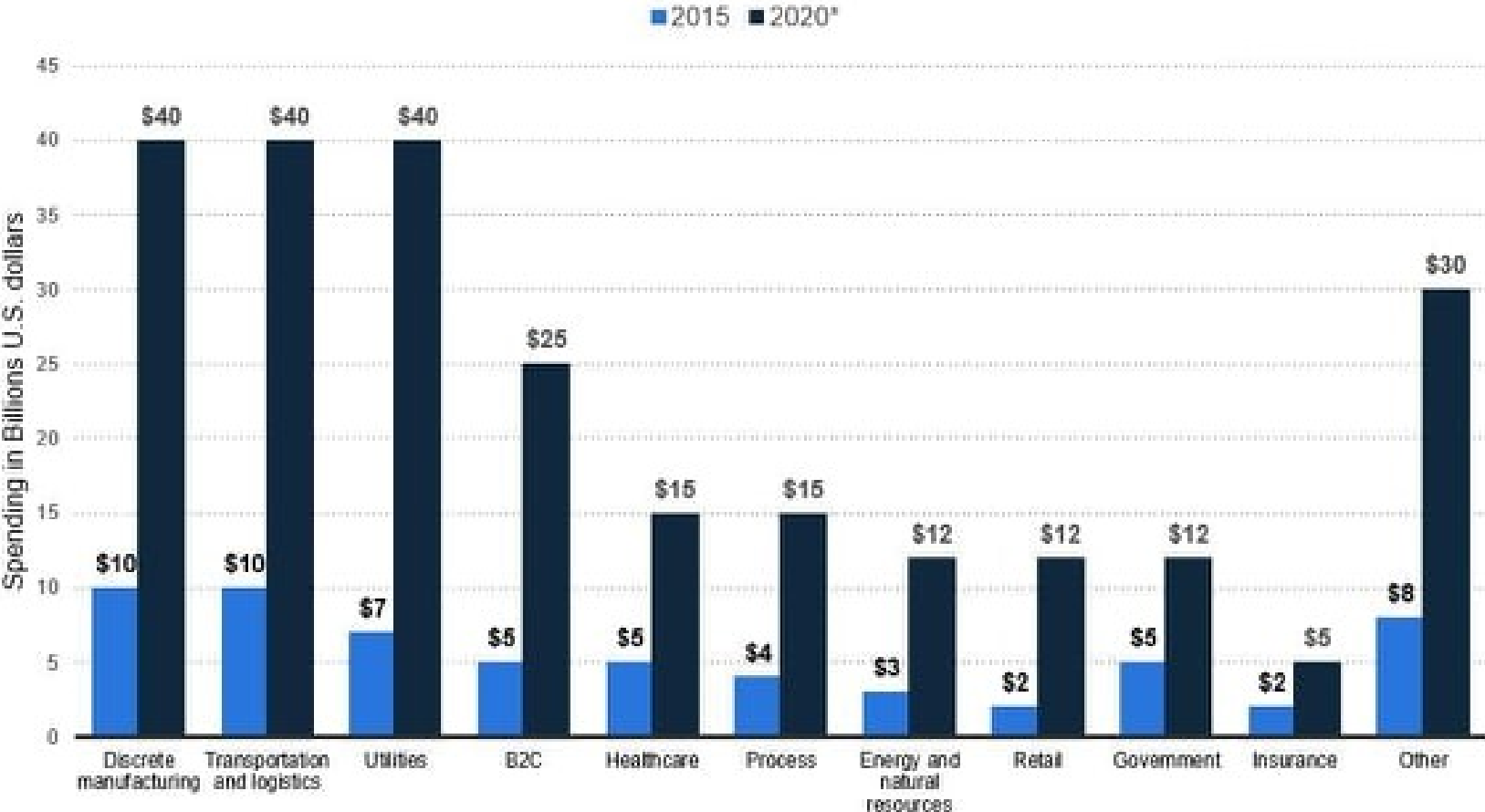
Potential impact of artificial intelligence (AI) on real gross value added (GVA) worldwide, by country, by 2035



Sources:
Accenture; Frontier Economics
© Statista 2017

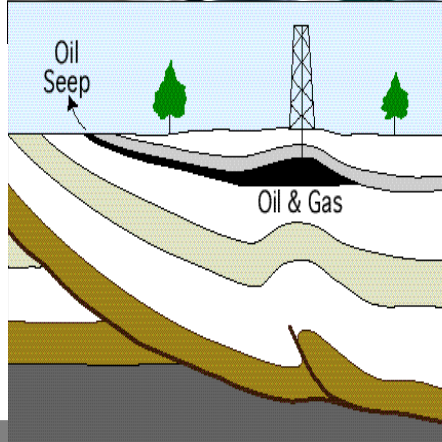
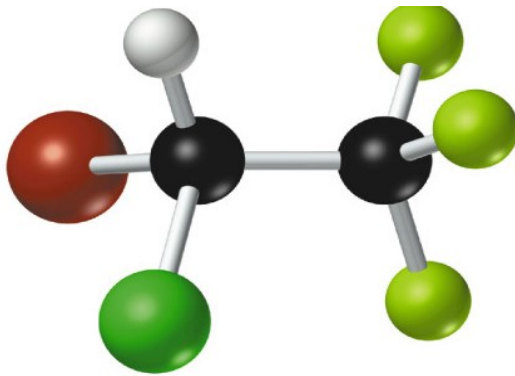
Additional Information:
Worldwide; 2016

Spending on Internet of Things Worldwide by Vertical in 2015 and 2020* (in billions of U.S. dollars)



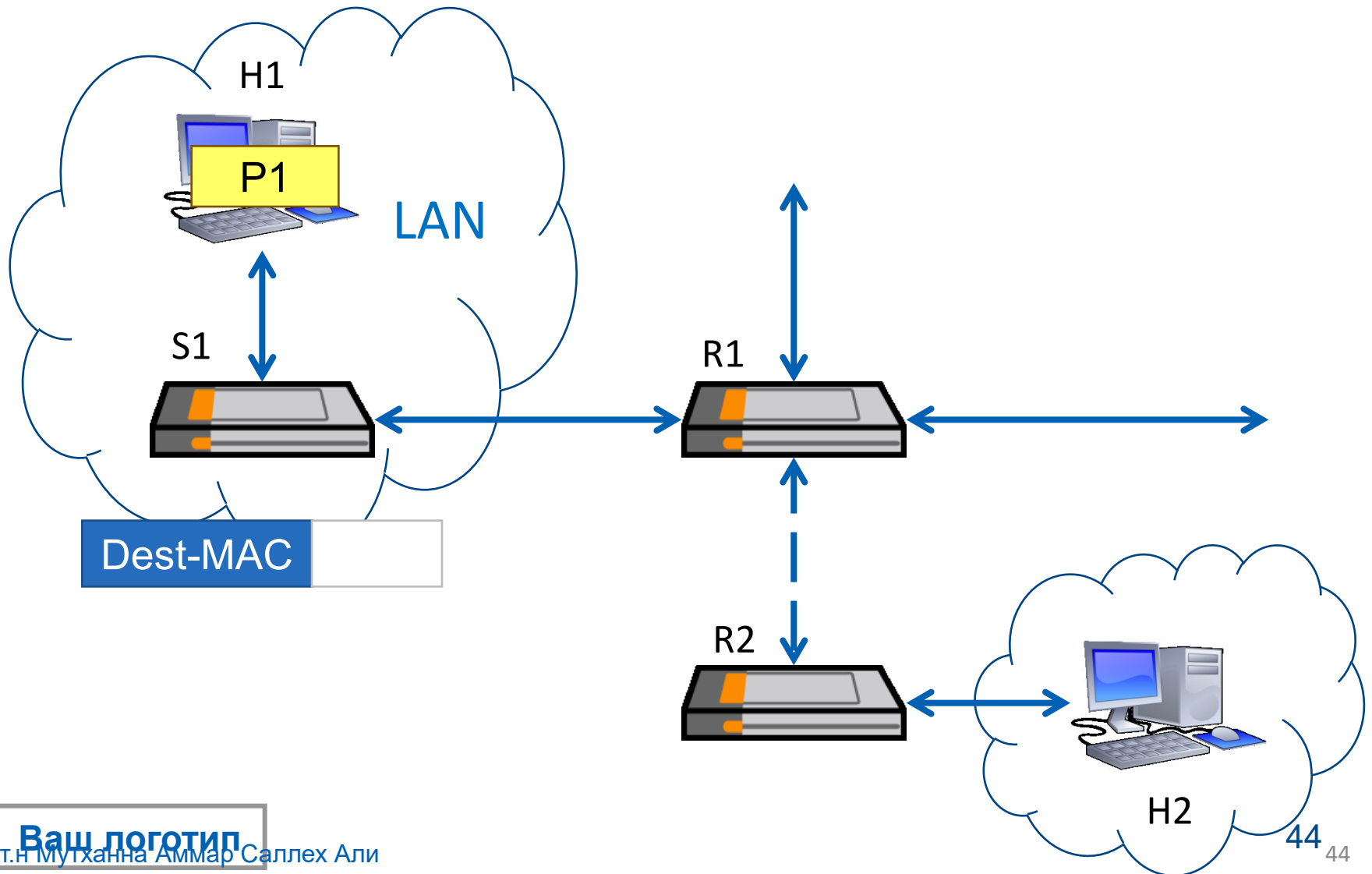
Примеры применения искусственного интеллекта

- Self Driving Cars
- Boston Dynamics
- Navigation Systems
- ASIMO
- Chatbots
- Human vs Computer Games
- Many More

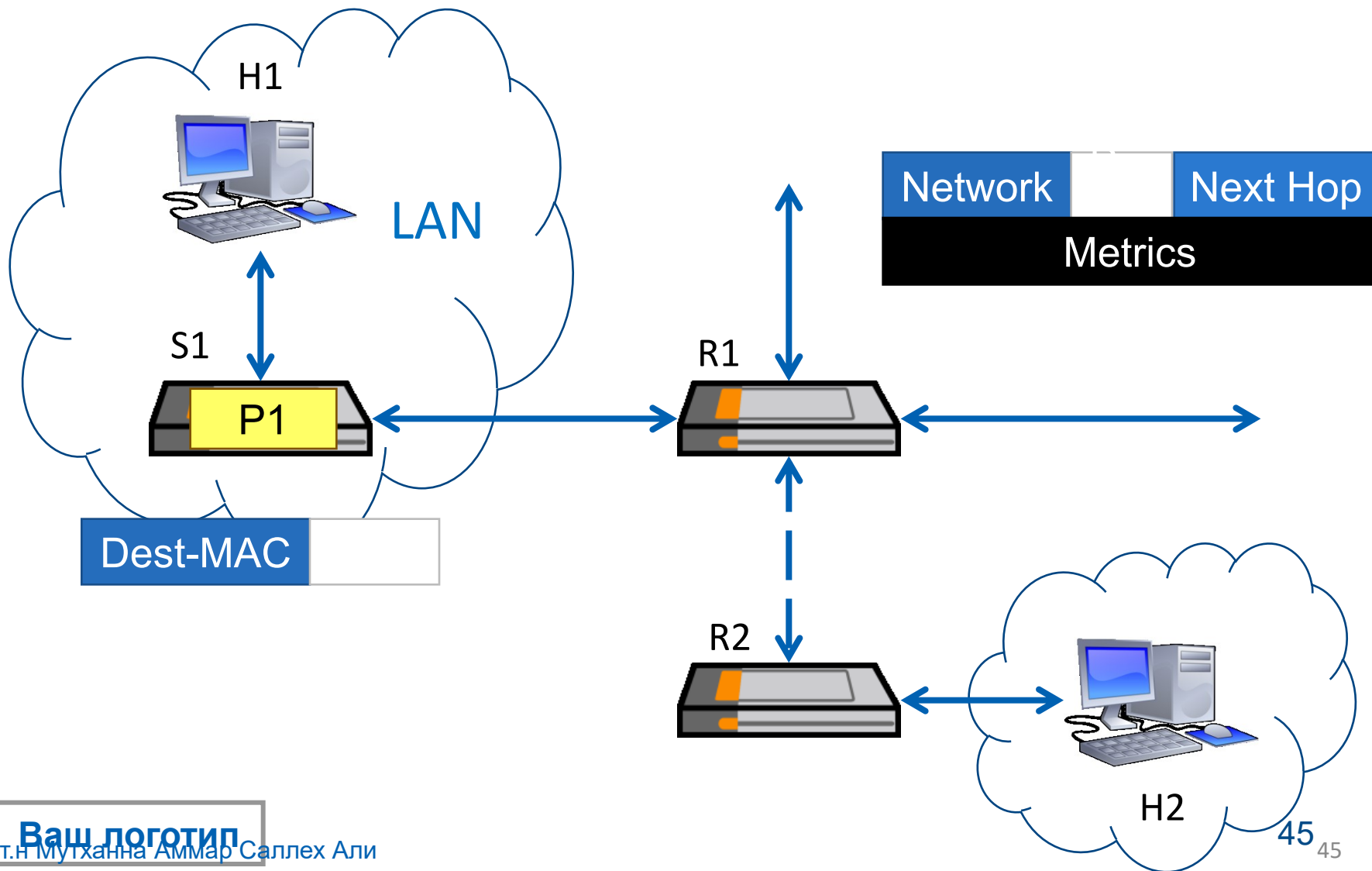


SDN

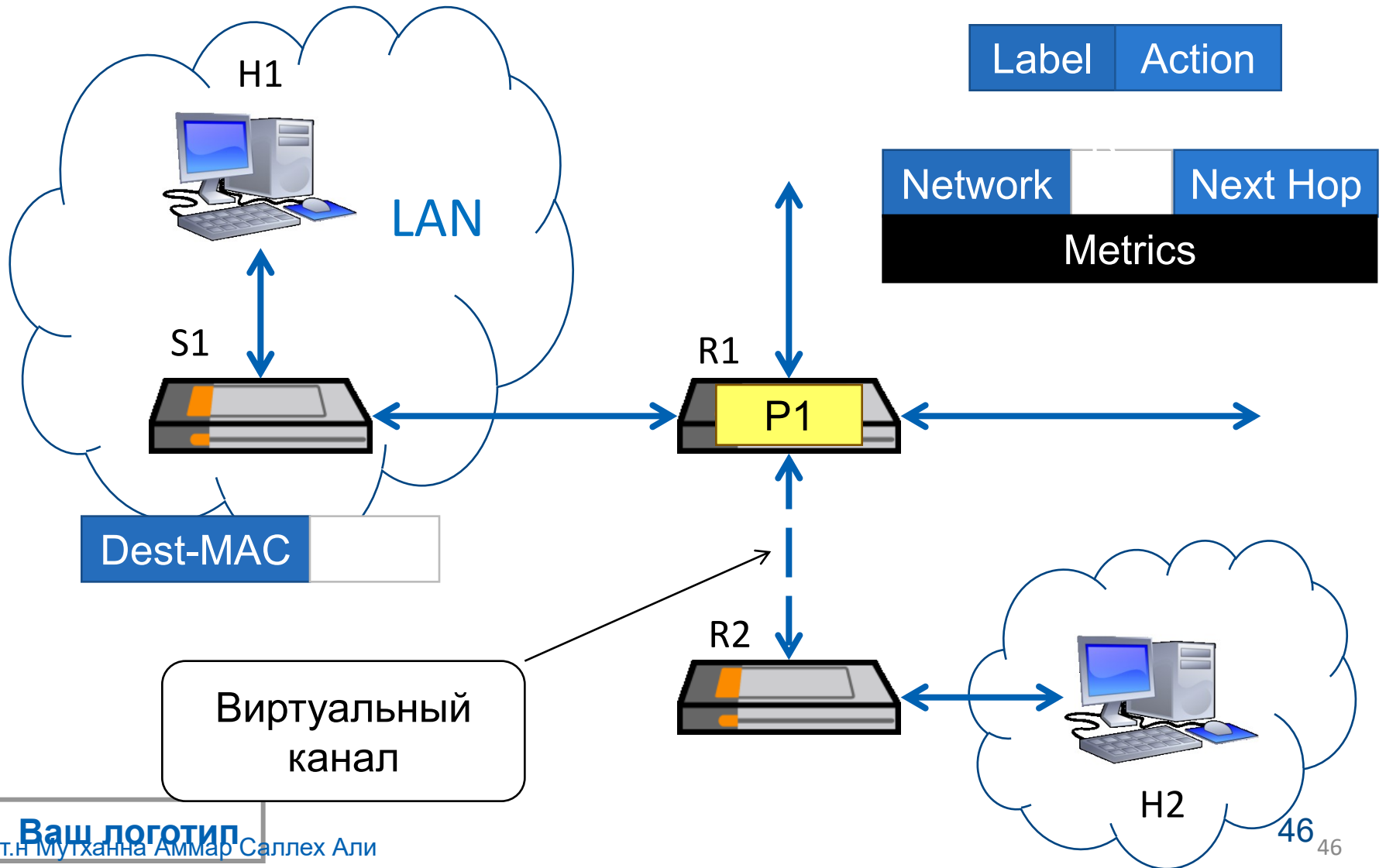
Коммутация пакетов



Коммутация пакетов



Коммутация меток



Программно-Конфигурируемые СЕТИ

[разделение передачи и управления]

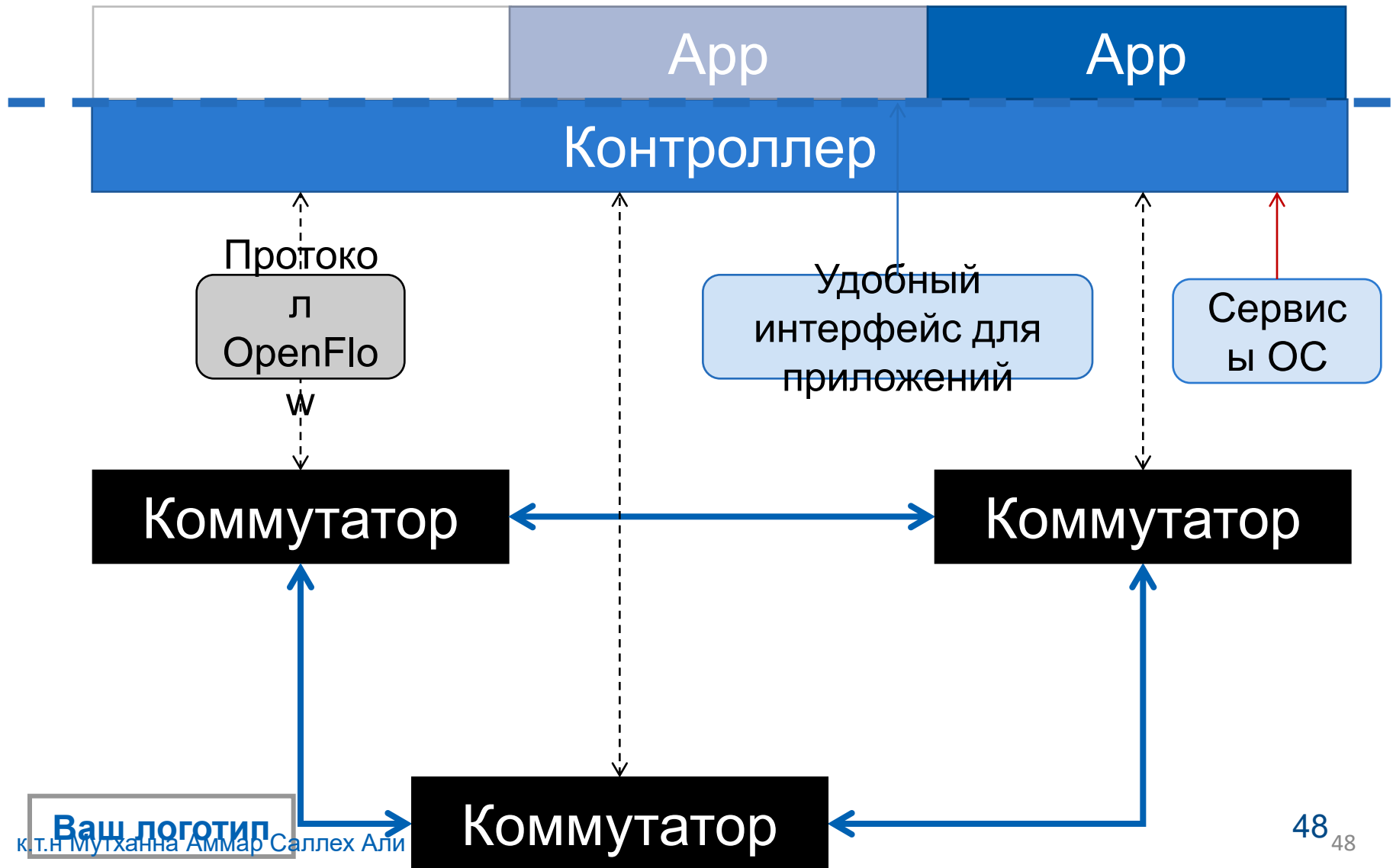
Управление / Control

Передача / Forwarding

- Существует фиксированный набор простых инструкций обработки пакетов
- Концепция совместима с различными протоколами
- Работает с разноуровневым оборудованием

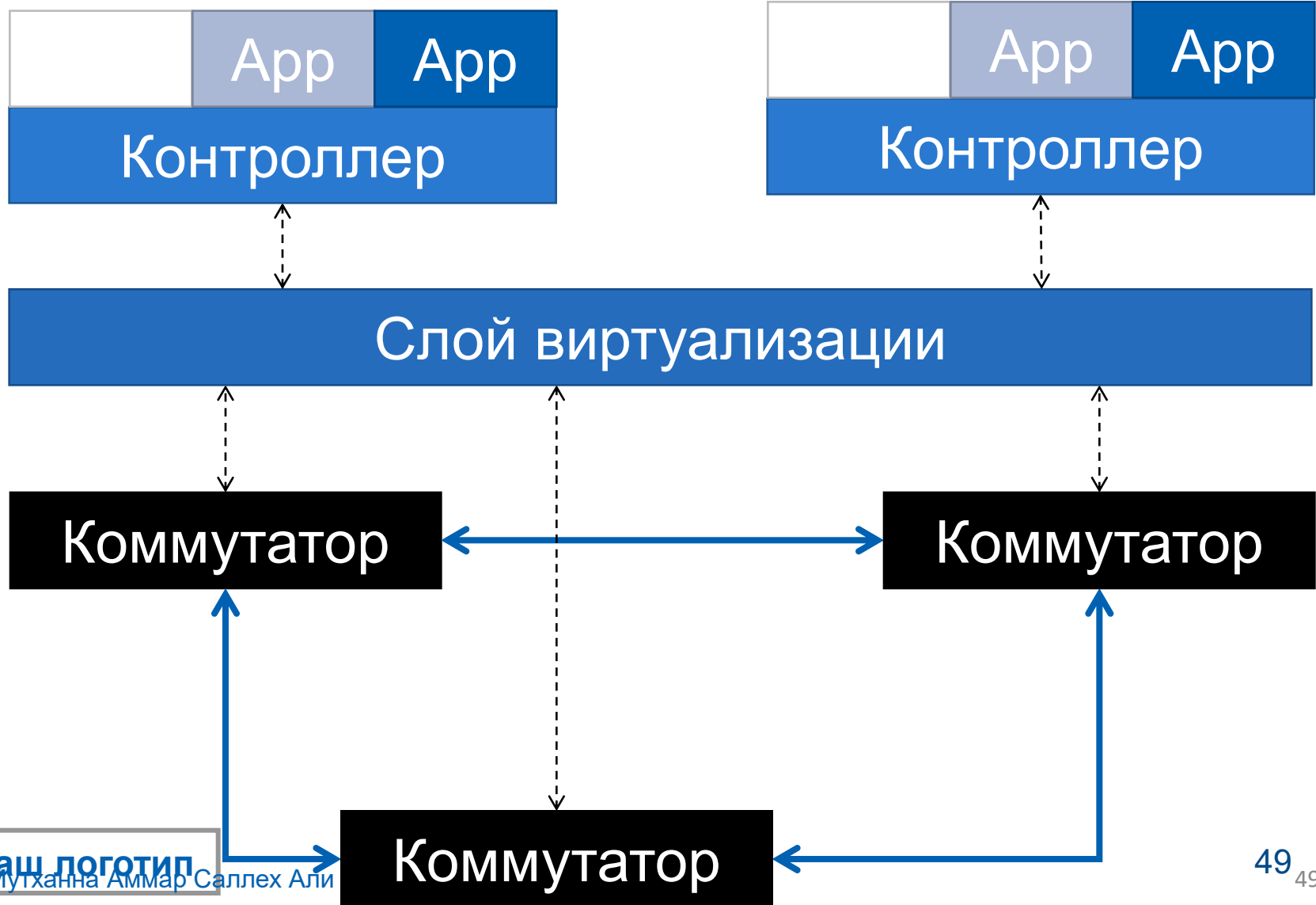
Программно-Конфигурируемые сети

[централизация управления]

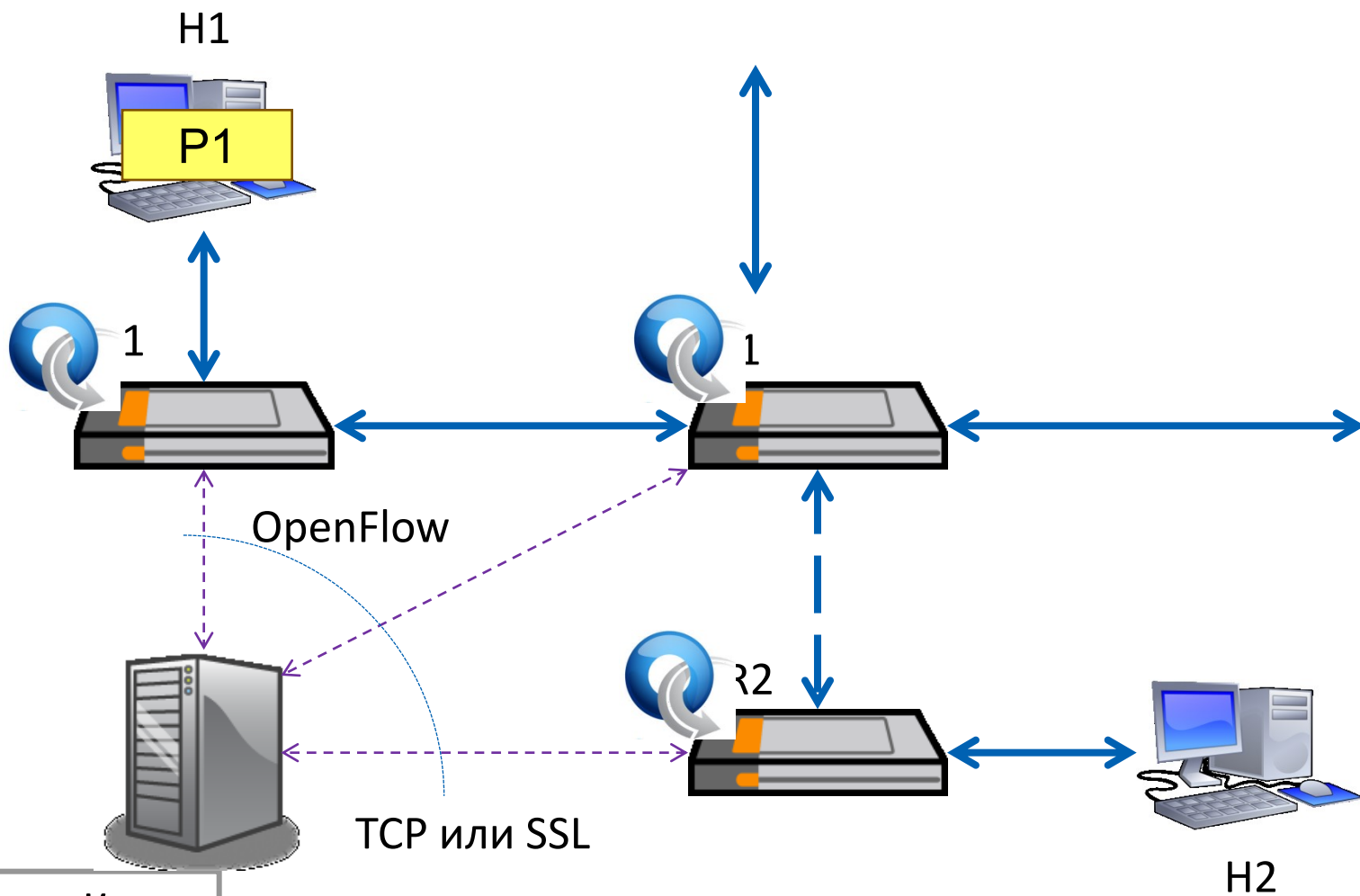


Программно-Конфигурируемые сети

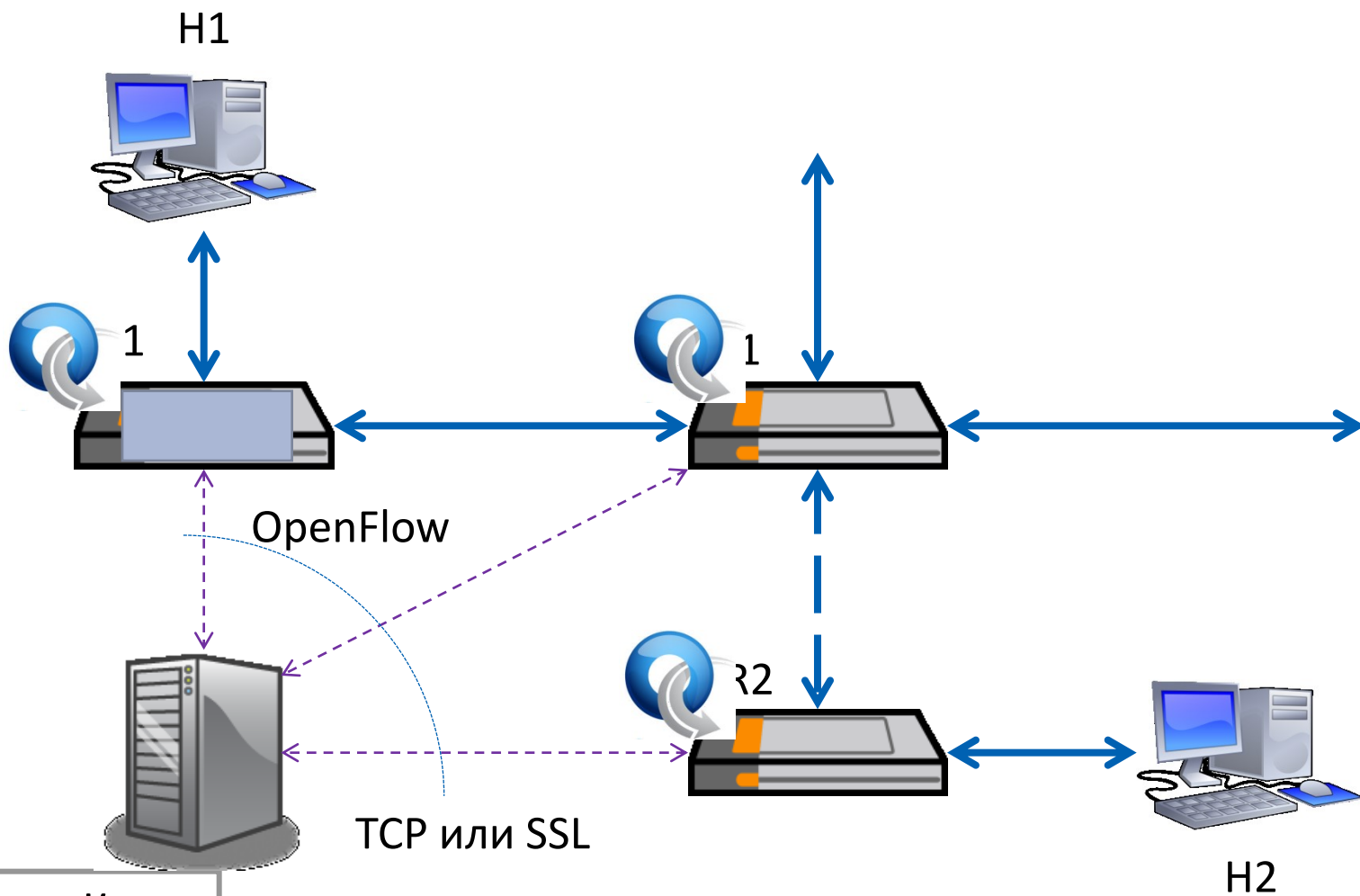
[виртуализация сети]



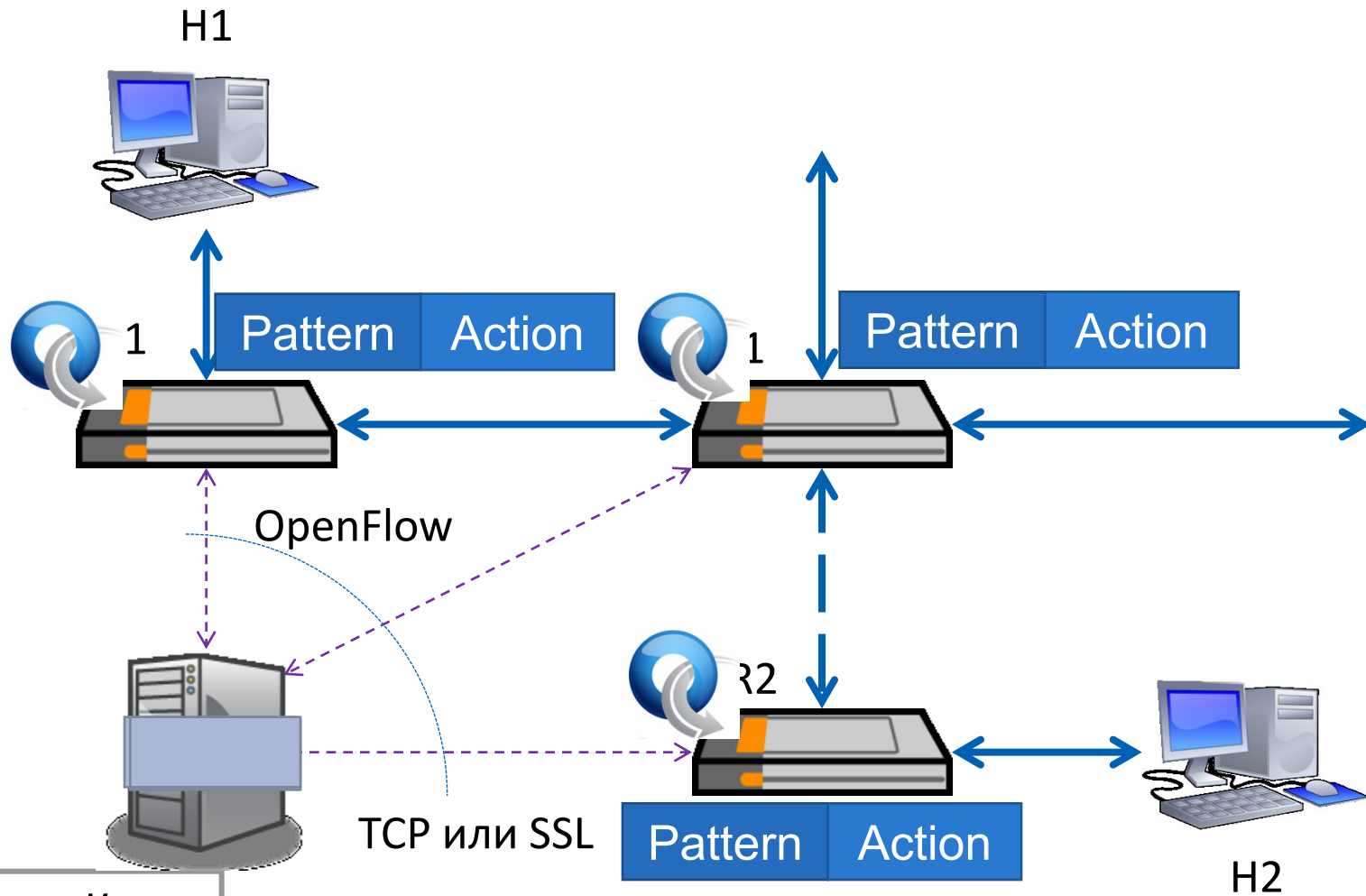
Сегмент ПКС



Сегмент ПКС

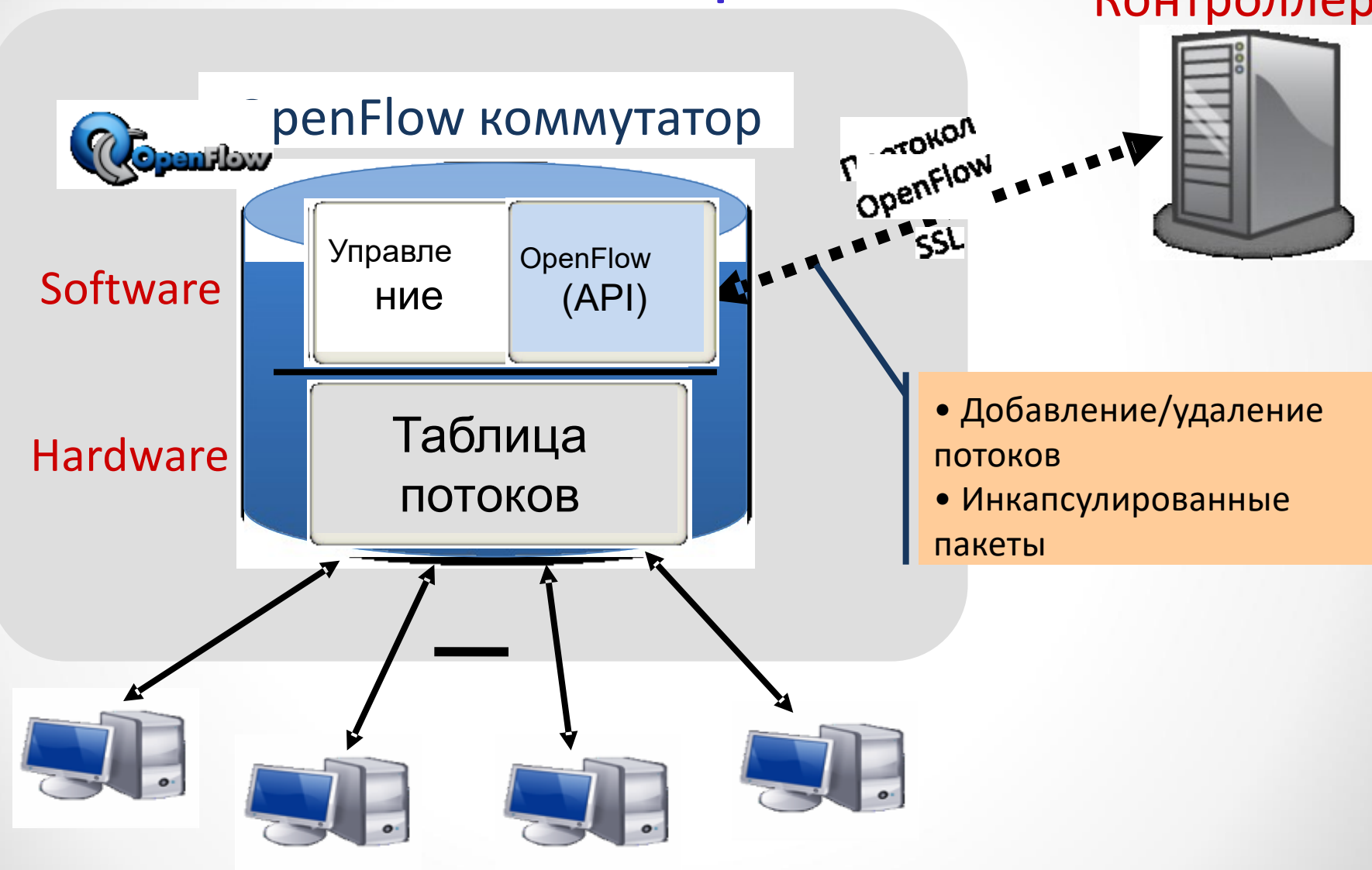


Сегмент ПКС



OpenFlow

Контроллер



OpenFlow протокол

Поддерживаются 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 Protocol Messages

Симметричные сообщения

Hello

Сообщение OFPT_HELLO не имеет тела; то есть, оно состоит только из заголовка OpenFlow. Должны быть подготовлены реализации, воспринимающие сообщения hello, которые имеют тело.

Запрос эхо

Сообщение *запроса эхо* (Echo Request) состоит из заголовка OpenFlow плюс поля данных произвольной длины. Поле данных может быть временной меткой сообщения для проверки задержки, или иметь нулевую длину при проверке работоспособности канала переключатель-контроллер.

Эхо отклик

Сообщение *эхо отклик* (Echo Reply) состоит из заголовка OpenFlow и немодифицированного поля данных эхо-запроса.

Сообщение об ошибке

Бывают моменты, когда switch вынужден оповестить контроллер о проблеме или наоборот. Это делается с помощью сообщения OFPT_ERROR.

Экспериментальные сообщения

OFPT_EXPERIMENTER Тип сообщений, которые нужны для различных функций (считывания баз, поддержки расширений и т.д.)

OpenFlow Protocol Messages

Сообщения контроллер-коммутатор

При установлении сессии контроллер посылает сообщение OFPT_FEATURES_REQUEST. Это сообщение не содержит тела после заголовка OpenFlow. Коммутатор откликается сообщением OFPT_FEATURES_REPLY.

Конфигурирование коммутатора

Контроллер способен установить конфигурацию и параметры очереди в коммутаторе с помощью сообщений OFPT_SET_CONFIG и OFPT_GET_CONFIG_REQUEST, соответственно.

Переключатель реагирует на запрос конфигурации сообщением OFPT_GET_CONFIG_REPLY; на запрос конфигурации отклика не посылается.

Асинхронные сообщения

Сообщение packet-In

Когда пакет получен матрицей коммутации (datapath) и послан контроллеру, используется сообщение OFPT_PACKET_IN.

OpenFlow Protocol Messages

Сообщение OFPT_FLOW_REMOVED

Если контроллер запросил получение уведомлений, когда поток записей завершен по тайм-ауту или удаляются из таблиц, то DataPath делает это с сообщением OFPT_FLOW_REMOVED.

Сообщение OFPT_PORT_STATUS

Если конфигурационные биты порта изменены через другой интерфейс коммутатора, то он посылает сообщение OFPT_PORT_STATUS, чтобы уведомить контроллер об этом изменении

Сообщение вывода

Когда контроллер хочет послать пакет к определенному порту data path, он использует сообщение OFPT_PACKET_OUT.

Сообщения изменения состояния

Модификации таблицы переадресации со стороны контроллера осуществляется с помощью сообщений OFPT_FLOW_MOD

OpenFlow Protocol Messages

Сообщения изменения состояния

Изменения в таблице групп из контроллера выполняются с сообщением OFPT_GROUP_MOD

Сообщение модификации порта

Контроллер для модификации поведения порта использует сообщение OFPT_PORT_MOD.

Сообщение модификации таблиц

Контроллер для модификации таблиц использует сообщение OFPT_TABLE_MOD

Составные сообщения

Во время работы системы, контроллер может запросить состояние Data Path с помощью OFPT_MULTIPART_REQUEST

Коммутатор отвечает одним или несколькими OFPT_MULTIPART_REPLY

Сообщения Barrier.

Когда контроллер хочет обеспечить выполнение всех зависимостей для сообщения или хочет получать уведомления о завершенных операциях, он может использовать сообщение OFPT_BARRIER_REQUEST. Это сообщение не имеет тела.

При получении, коммутатор должен завершить обработку всех ранее принятых сообщений, в том числе отправка соответствующего ответа или сообщения об ошибках, перед выполнением действий с сообщением на запрос Barrier. Когда такая обработка завершена, коммутатор должен послать сообщение OFPT_BARRIER_REPLY с xID первоначального запроса.

OpenFlow Protocol Messages

Сообщения конфигурации очередей

OFPT_QUEUE_GET_CONFIG_REQUEST и
OFPT_QUEUE_GET_CONFIG_REPLY

Set Asynchronous Configuration Message

Контроллер может устанавливать и запрашивать асинхронные сообщения, которые он хочет получить (кроме сообщений об ошибках) на данном канале OpenFlow при помощи OFPT_SET_ASYNC и OFPT_GET_ASYNC_REQUEST, соответственно.

Коммутатор отвечает на сообщение OFPT_GET_ASYNC_REQUEST сообщением OFPT_GET_ASYNC_REPLY; он не отвечает на запрос, по установке конфигурации.

Сообщения конфигурации счетчика

Изменения в счетчике от контроллера выполняются при помощи сообщения OFPT_METER_MOD

Пример подключения коммутатора

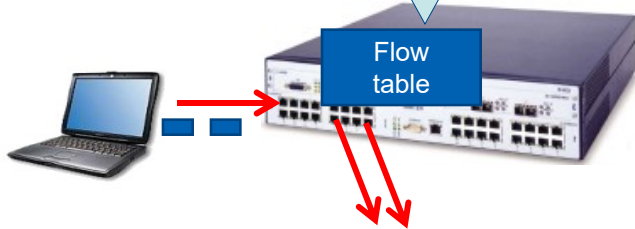


OpenFlow: Flow Table

Flow Table

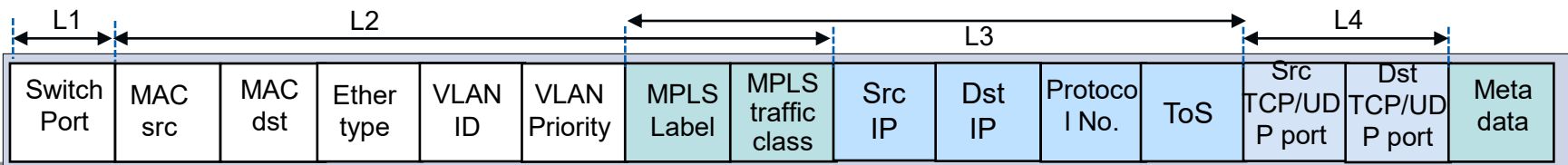
Счетчики
используются,
когда контроллер
вычисляет пути

Flow entry	match field	counter	Action (Instruction)	priority	Timeout	cookie
1						
n				



- Actions(Instructions)**
1. Forward packet to port(s)
 2. Encapsulate and forward to controller
 3. Drop packet
 4. Send to normal processing pipeline
 5. Modify Fields
 6. Etc.

- Match field= L1~L4 header information
 - OpenFlow 1.0 → 12 tuples
 - OpenFlow 1.1 → 15 tuples
 - OpenFlow 1.3 → 40 tuples (158 bytes)



Match fields of OpenFlow 1.1

OpenFlow: Flow Table

■ Flow Table

Operation Mode	Switch Port	MAC src	MAC dst	Ether type	VLAN ID	Src IP	Dst IP	Proto No.	TCP S_port	TCP D_port	Action	Counter
Switching	*	*	00:1f..	*	*	*	*	*	*	*	Port1	243
Flow Switching	Port3	00:20..	00:2f..	0800	vlan1	1.2.3.4	1.2.3.9	4	4666	80	Port7	123
Routing	*	*	*	*	*	*	1.2.3.4	*	*	*	Port6	452
VLAN Switching	*	*	00:3f..	*	vlan2	*	*	*	*	*	Port6 Port7 Port8	2341
Firewall	*	*	*	*	*	*	*	*	*	22	Drop	544
Default Route	*	*	*	*	*	*	*	*	*	*	Port1	1364

Packet Forwarding in OpenFlow

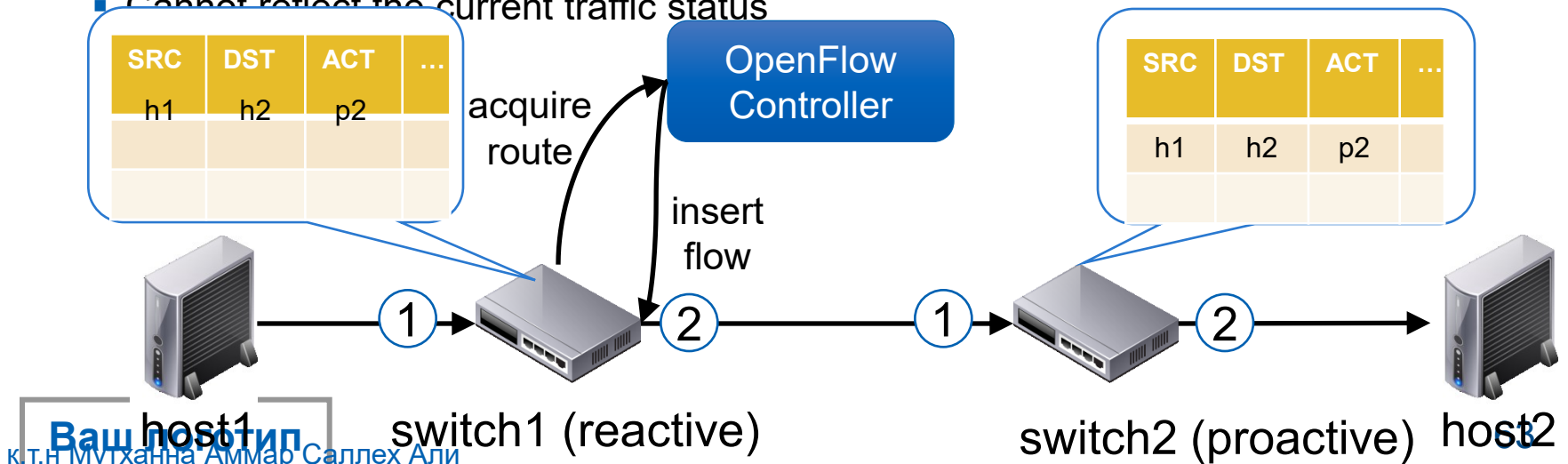
■ Packet Forwarding

■ Reactive flow insertion

- A non-matched packet reaches to OpenFlow switch, it is sent to the controller, based on the info in packet header, an appropriate flow will be inserted
- Always need to query the path from controller during packet arrival → slow
- Can reflect the current traffic status

■ Proactive flow insertion

- Flow can be inserted proactively by the controller to switches before packet arrives
- No need to communicate during packet arrival → fast packet forwarding
- Cannot reflect the current traffic status



Исследование топологии сети в OpenFlow

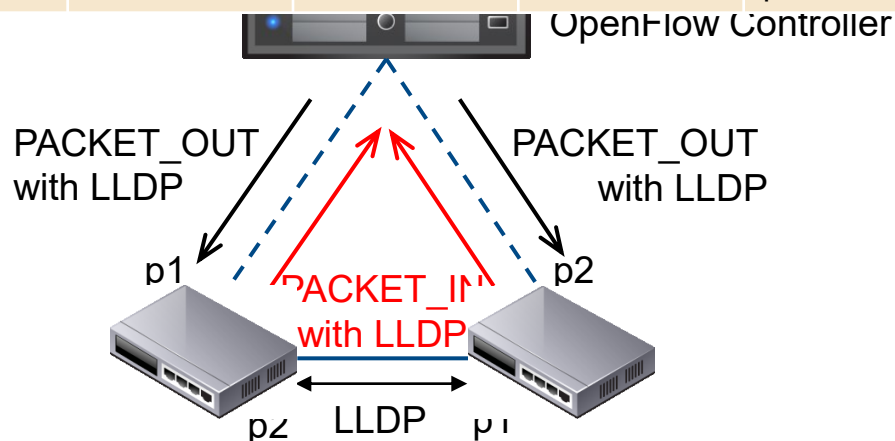
Цель

- Для построения архитектуры сети

Метод

- Использование Link Layer Discovery Protocol (LLDP)

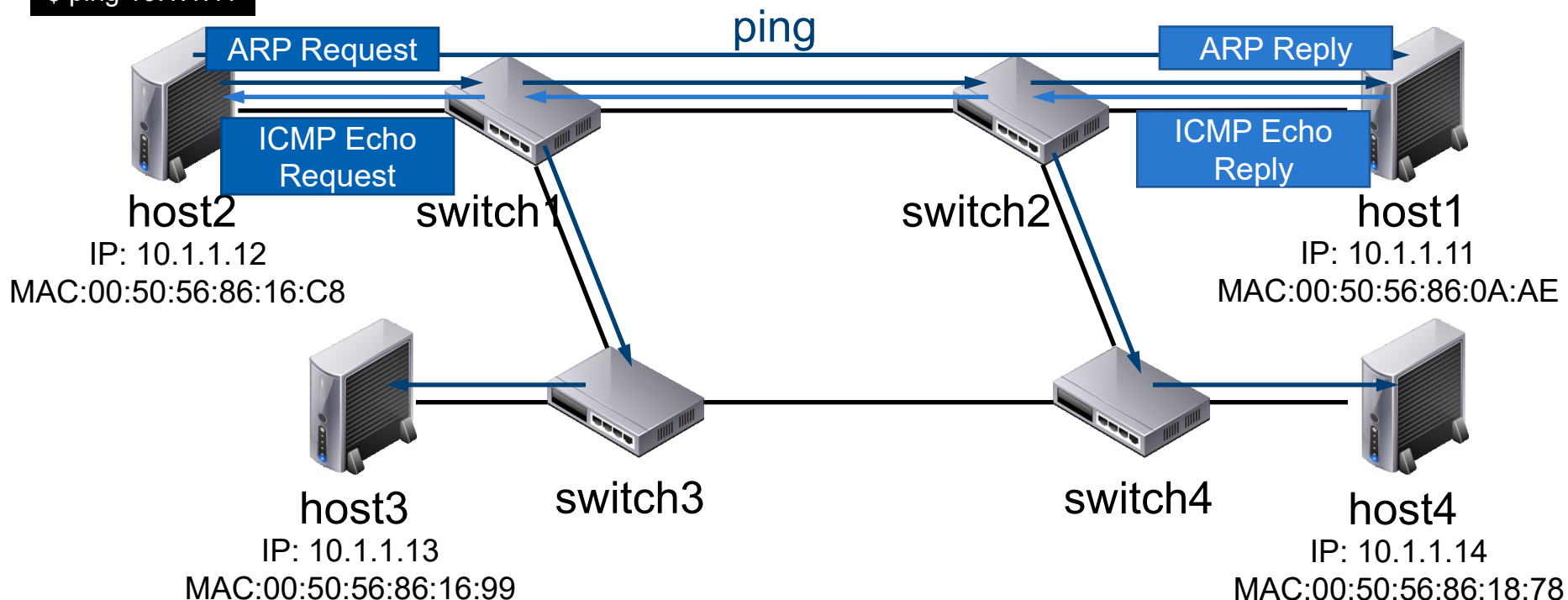
IDX	SRC	DST	SRC PORT	DST PORT
153	sw. A	sw. B	p2	p1
...
357	sw. B	sw. A	P1	p2



Взаимодействие в традиционной сети

1. host2 tries communication to host1 by sending a ping ICMP packet
2. host2 broadcasts ARP Request packet
3. host1 replies ARP Request with ARP Reply
4. host2 creates entry to ARP Cache Table
5. host2 sends ICMP Echo request packet
6. host1 replies ICMP Echo request with ICMP Echo reply

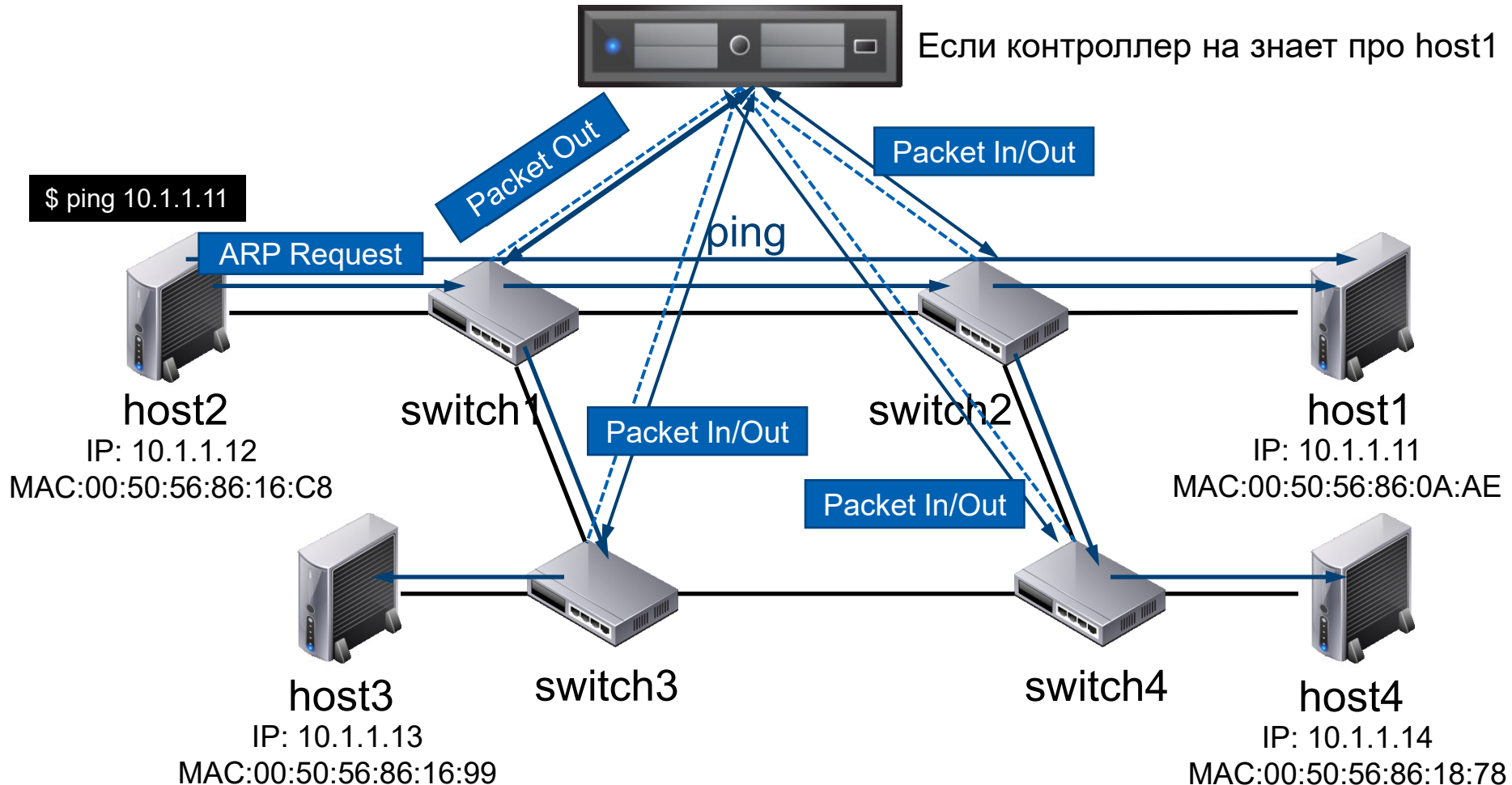
\$ ping 10.1.1.11



ARP Cache Table of Host2

Internet Address	Physical Address	Type
10.1.1.254	00-00-0C-E7-58-CD	Dynamic
10.1.1.11	00-50-56-86-0A-AE	Dynamic

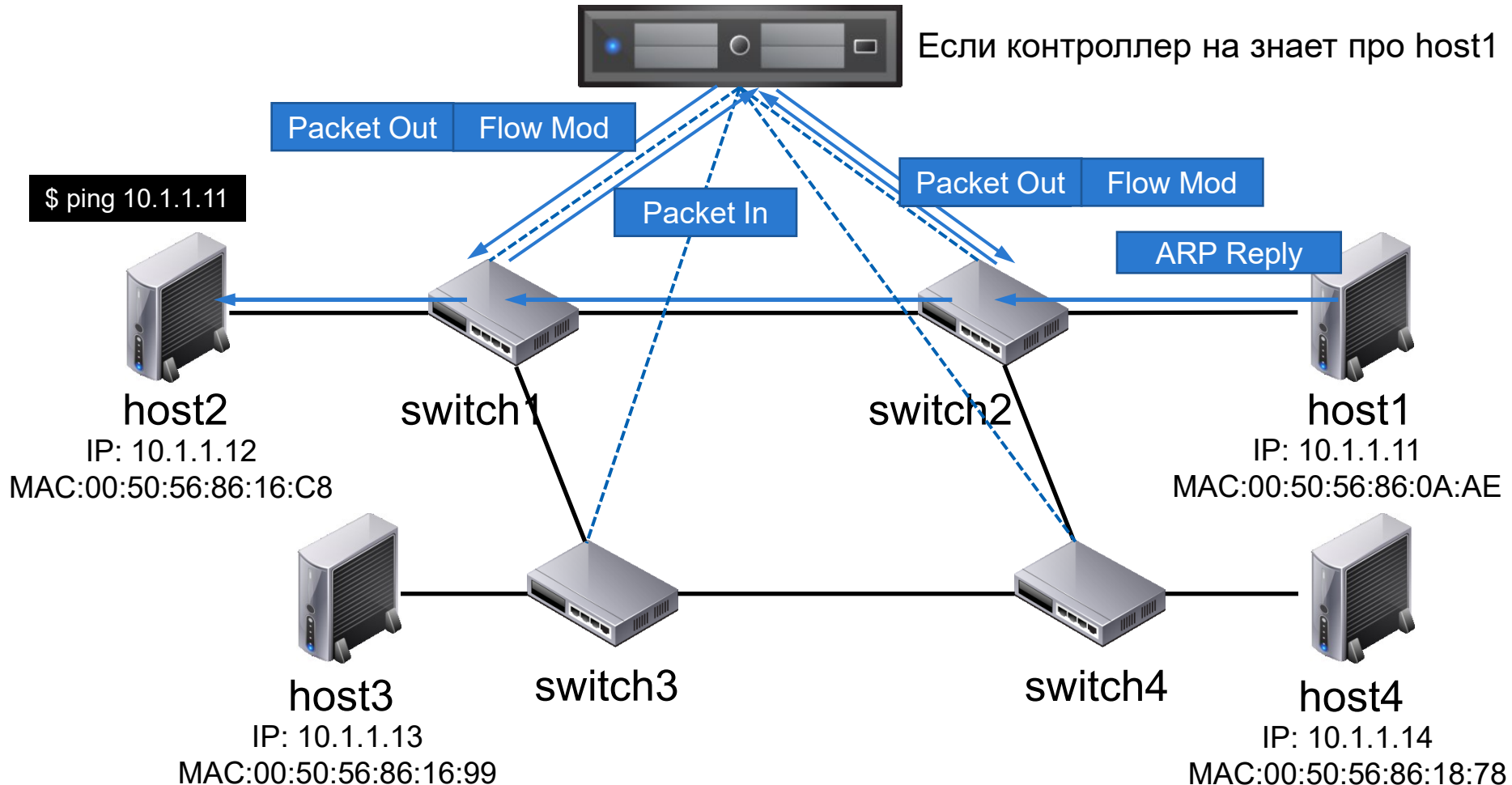
Взаимодействие в OpenFlow



ARP Cache Table of Host2

Internet Address	Physical Address	Type
10.1.1.254	00-00-0C-E7-58-CD	Dynamic

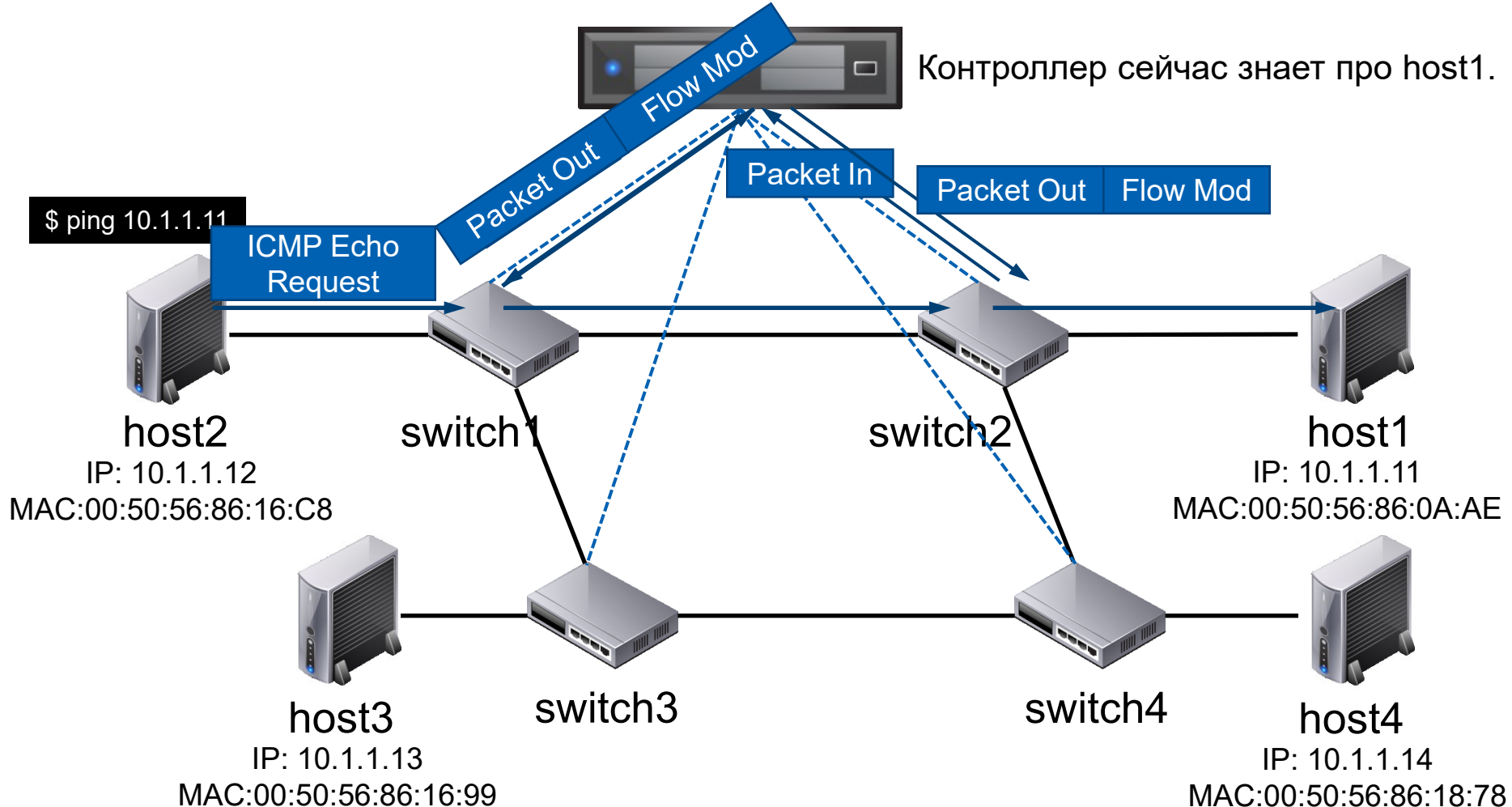
Взаимодействие в OpenFlow



ARP Cache Table of Host2

Internet Address	Physical Address	Type
10.1.1.254	00-00-0C-E7-58-CD	Dynamic
10.1.1.11	00-50-56-86-0A-AE	Dynamic

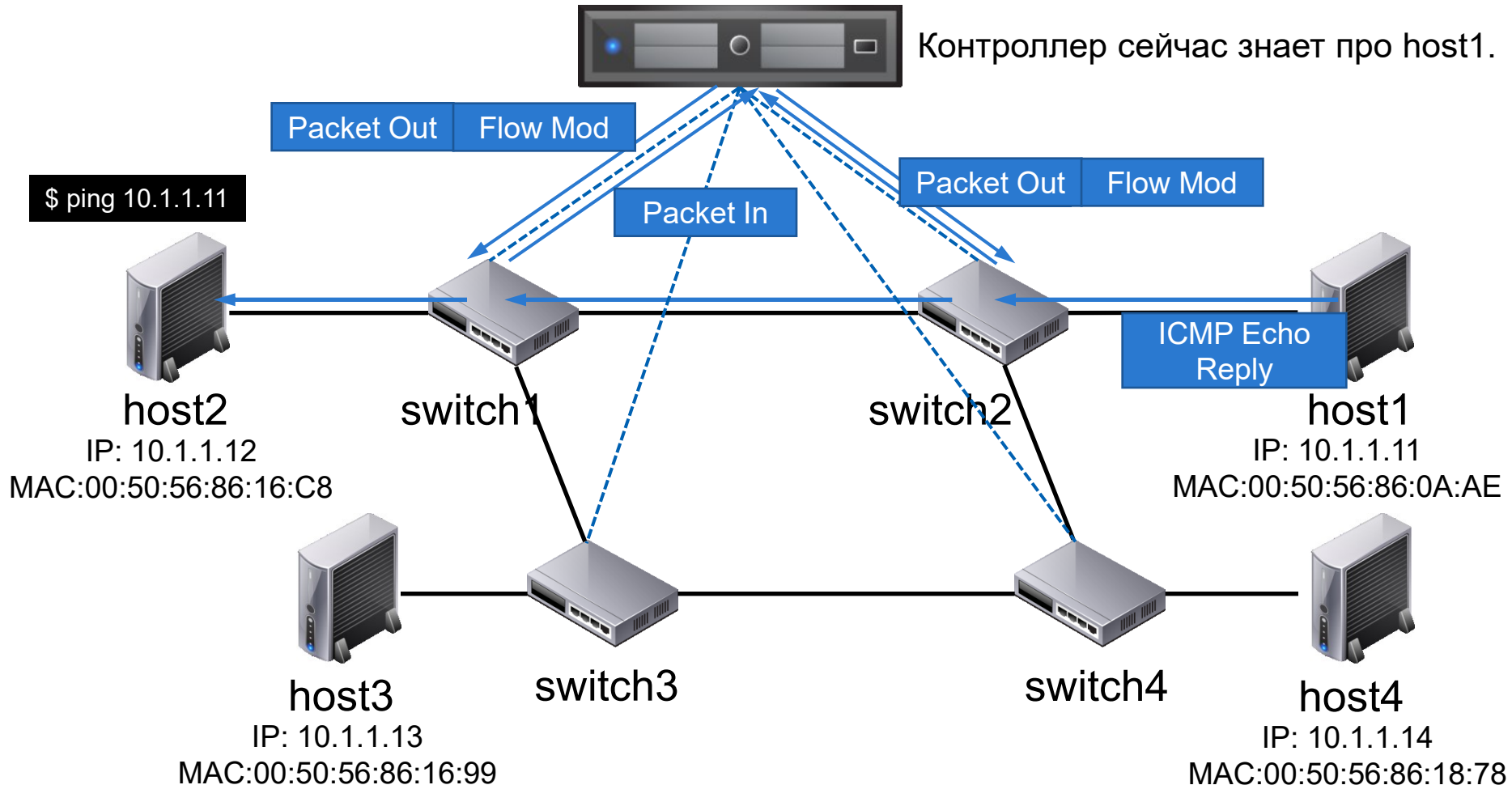
Заимодействие в OpenFlow



ARP Cache Table of Host2

Internet Address	Physical Address	Type
10.1.1.254	00-00-0C-E7-58-CD	Dynamic
10.1.1.11	00-50-56-86-0A-AE	Dynamic

Взаимодействие в OpenFlow



ARP Cache Table of Host2

Internet Address	Physical Address	Type
10.1.1.254	00-00-0C-E7-58-CD	Dynamic
10.1.1.11	00-50-56-86-0A-AE	Dynamic

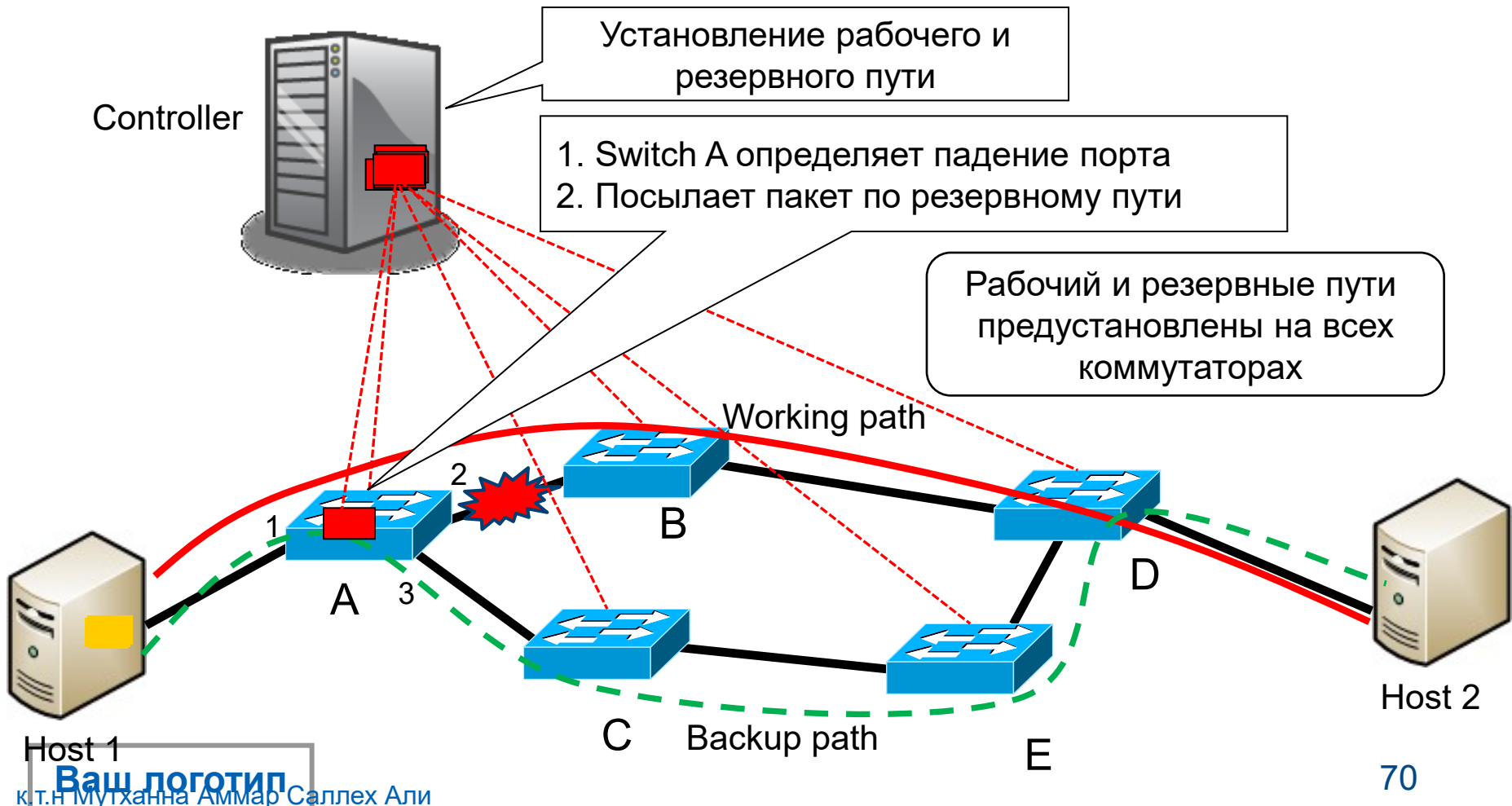
OpenFlow Отказоустойчивость

OpenFlow Failover

- Защита

Flow table of Switch A (group table combined)

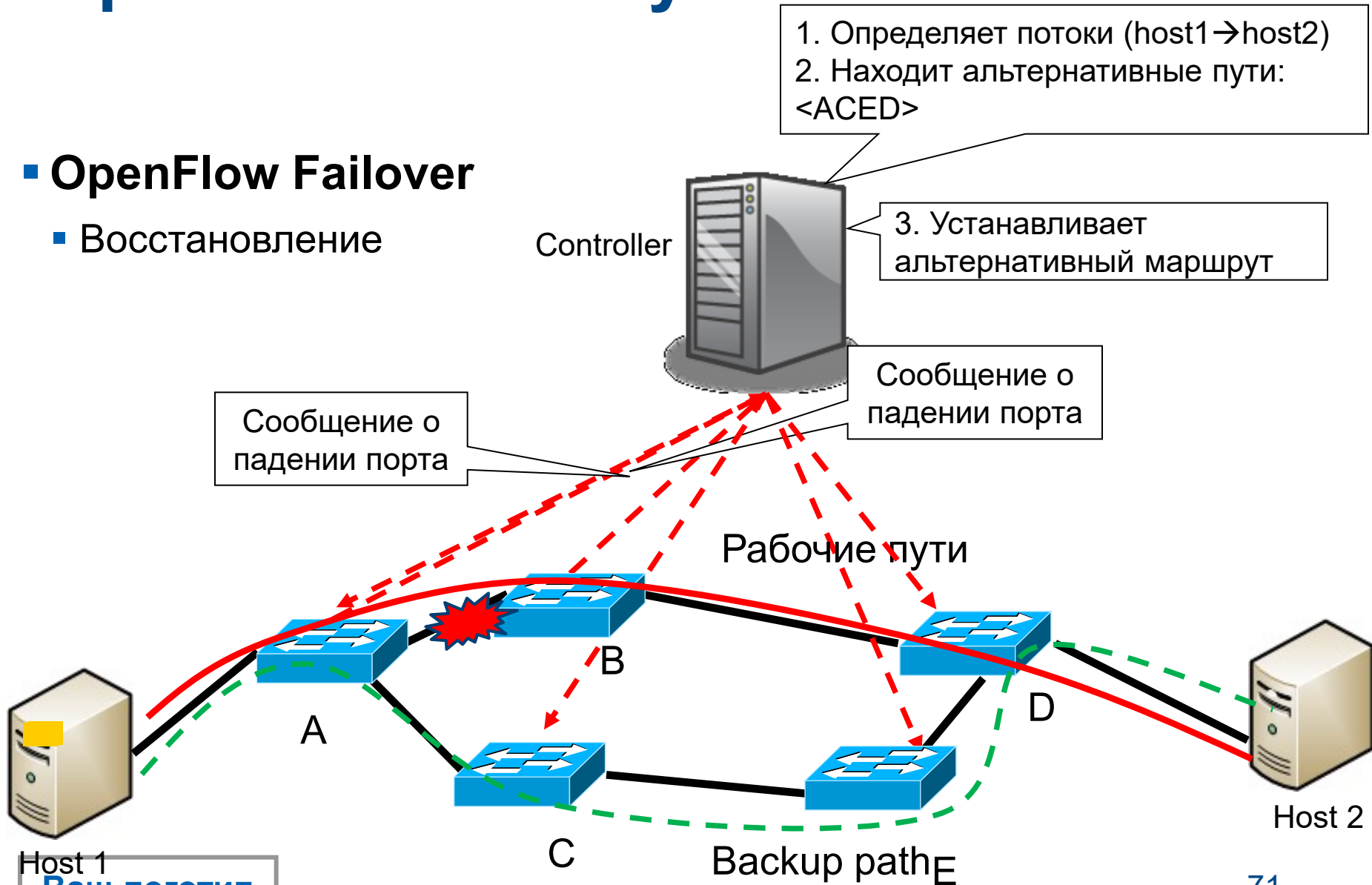
src	dst	Out port	Failover port
h1	h2	2	3



OpenFlow Отказоустойчивость

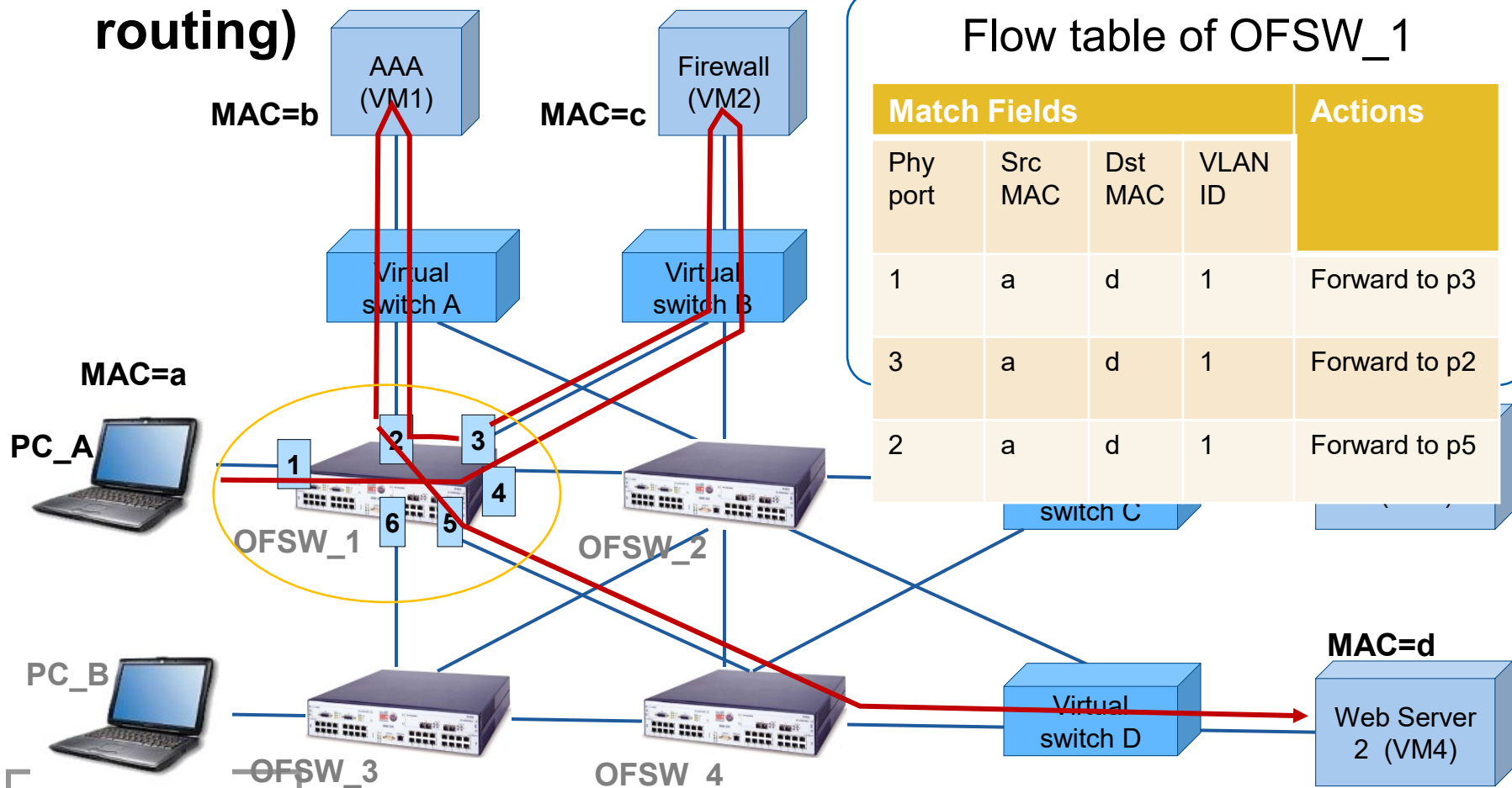
OpenFlow Failover

- Восстановление



OpenFlow Пример

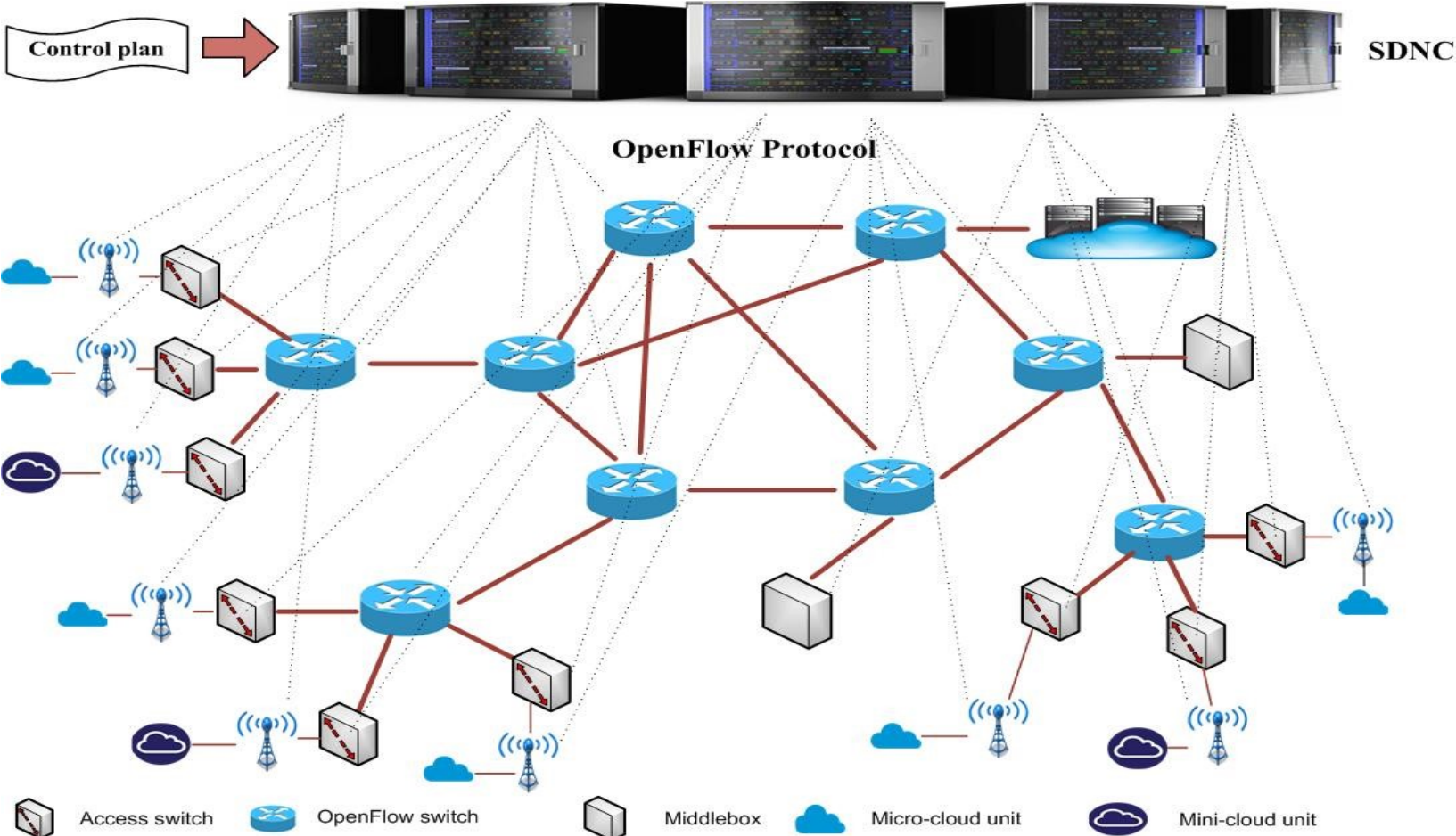
- Пример управления маршрутизации (hop-by-hop routing)



Flow table of OFSW_1

Match Fields				Actions
Phy port	Src MAC	Dst MAC	VLAN ID	
1	a	d	1	Forward to p3
3	a	d	1	Forward to p2
2	a	d	1	Forward to p5

Интеллектуальное ядро для сетей 5G



Многоуровневая облачная архитектура для сетей IMT-2020/5G с ультра малыми задержками

Предложенная архитектура отличается ранее известных введением дополнительного уровня в иерархии. В нескольких источниках ранее предлагалось использовать модели, которые условно можно назвать трехуровневыми: помимо публичной облачной платформы и основного облака сети, в таких моделях предлагался только один уровень периферийных облаков (тучек или мини-облаков), в то время как в рассматриваемой модели периферийный уровень также предлагается разбить на два уровня иерархии – микро-облака и мини-облака.

Многоуровневая облачная архитектура для сетей IMT-2020/5G с ультра малыми задержками

Ваш логотип

к.т.н Муханна Аммар Саллех Али

Многоуровневая облачная архитектура для сетей IMT-2020/5G с ультра малыми задержками



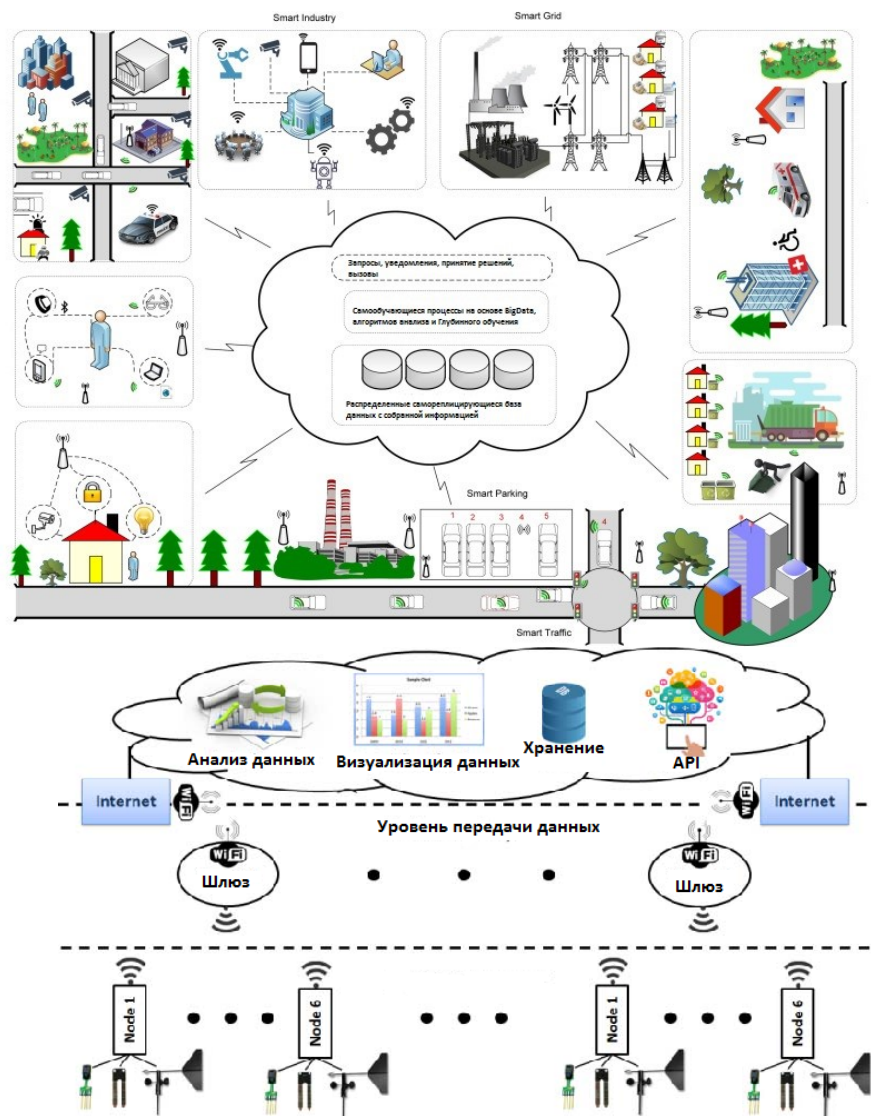
Наши проекты ИИ в сетях связи

Роль IoT в концепции Умного города

Концепция Умного города:

- В социальном плане: концепция по интеграции информационных и коммуникационных технологий, которая призвана повысить уровень комфорта, скорость и эффективность взаимодействия городских служб, снизить потребление ресурсов.
- В технологической плане: сложная взаимосвязанная информационно-аналитическая система с применением таких концепций как IoT, BigData, и искусственного интеллекта

Устройства IoT такие как контроллеры, сенсоры, датчики и микрокомпьютеры, позволяют собирать определенную информацию, агрегировать ее и отправлять в ЦОД для дальнейшего анализа и принятия решений.

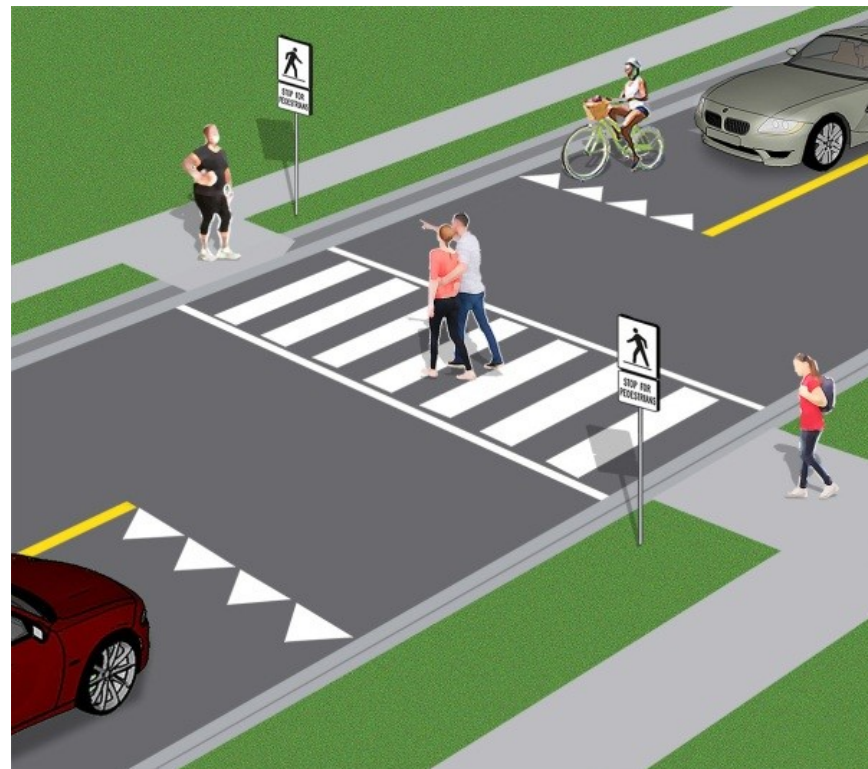


IoT в пешеходной системе Умного города

Одним из участников транспортной системы и дорожного движения является пешеход.

Главными задачами пешеходной системы Умного города является:

- Автоматическое распознавание пешеходов, намеревающихся перейти проезжую часть
- Отслеживание перемещения пешеходов во время пересечения проезжей части
- Подсчет числа пешеходов во время перехода проезжей части
- Информирование водителей о приближении к пешеходному участку
- Автоматическая подсветка перехода при обнаружении пешеходов
- Связь с другими умными устройствами транспортной системы и ЦОД



Определение пешеходов с применением сенсоров

Преимущества:

- Относительная простота разворачиваемой системы
- Низкая стоимость системы
- Поддержка современными датчиками многофункциональных опций по обнаружению объектов
- Возможность программирования сенсоров
- Возможность организовывать сеть из сенсоров, наращивая и улучшая процессы обнаружения

Недостатки:

- Большое число ложных срабатываний
- Невозможность определить тип пешеходов (намеревающегося перейти проезжую часть от простого пешехода)
- Отсутствие визуального подтверждения
- Невозможность определить пешеходный переход (только границы по микроволнам или иным специальным маркерам)

Определение пешеходов с применением компьютерного зрения

- **Централизованный подход**

Передачи потокового несжатого видео в облако для его дальнейшего анализа в режиме реального времени или постфактум.

- **Оконечный/узловой подход**

Анализ изображения и обнаружения объекта происходят на конечном узле. Сложность и скорость обнаружения требуемых объектов зависят от вычислительной мощности узла.



- **Гибридный подход**

Использование нейросетей конечными устройствами для обнаружения объекта. Обучение моделей для обнаружений с использованием Machine Learning/Deep Learning на мощных GPU-кластерах в ЦОДах. Постоянное обучение моделей за счет пополнения набора данных от узлов к центру.

Высокоуровневое проектирование системы

Основные шаги концепта системы:

- Распознавание пешехода, находящегося определенное время в непосредственной близости от «зебры»
- Запуск процедуры перехода проезжей части
- Повтор данных шагов через определенный интервал времени

Основные преимущества Asus Tinker Board:

- четырёхъядерный 1,8 ГГц ARM Cortex-A17
- GPU с поддержкой OpenGL ES1.1/2.0/3.0, OpenVG1.1, OpenCL, DirectX11
- 2 ГБ LPDDR3
- Wi-Fi: 802.11 b/g/n, LAN, Bluetooth
- GPIO разъем, разъём Camera Serial Interface

Требования к системе:

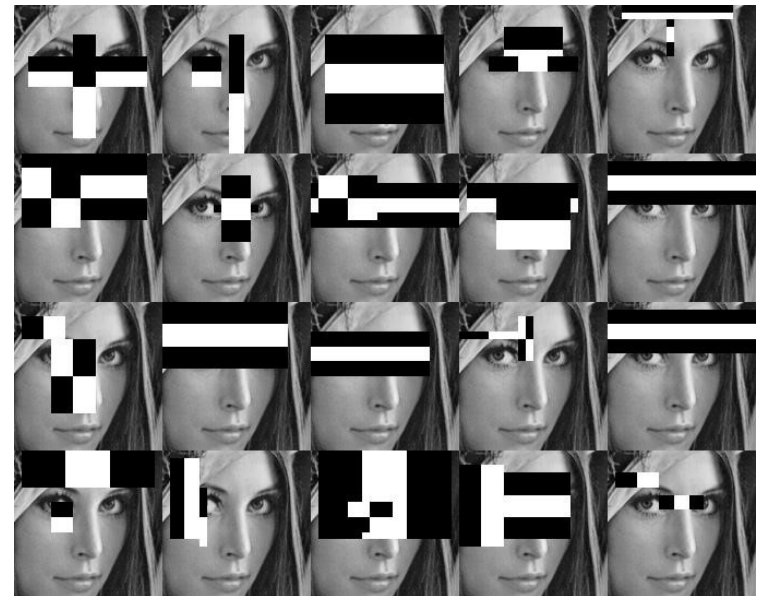
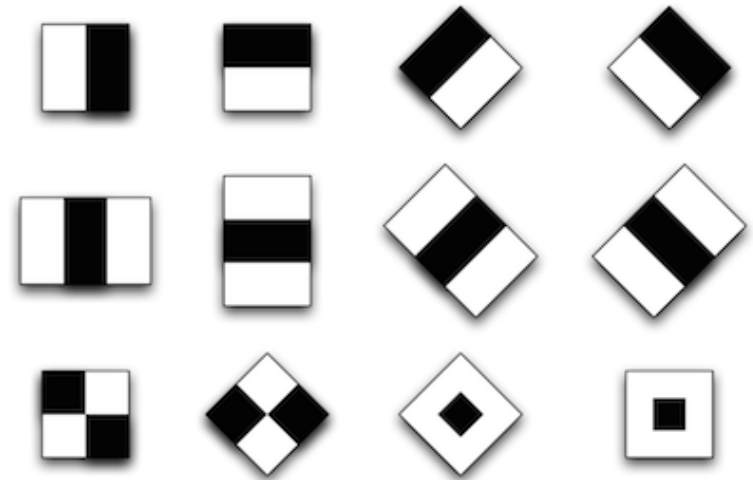
- мобильность
- обнаружение на конечном устройстве
- поддержка приемлемой скорости обнаружения
- низкое энергопотребление системы
- возможность «общаться» с другими узлами или подобными устройствами



Классическая модель распознавания

Одним из основополагающим способов является метод Виолы-Джонс («Accepted Conference On Computer Vision And Pattern Recognition» 2001 г)

- изображения конвертируются в интегральном представлении
- использование признаков Хаара
- использование бустинга для выбора наиболее подходящих признаков
- все признаки поступают на вход классификатора, который даёт результат «верно»/«ложь»
- использование каскада признаков для быстрого отбрасывания окон, где не найден объект



В процессе поиска, вычисление всех признаков занимает огромные вычислительные ресурсы и предназначен для запуска только на мощных компьютерах, но никак не на микрокомпьютерах и устройствах IoT. Классификатор должен реагировать только на определенное, нужное подмножество всех признаков. Отсюда следует, что надо обучить классификатор нахождению лиц по данному определенному подмножеству. Это можно сделать, обучая вычислительную машину автоматически с использованием технологий машинного обучения.

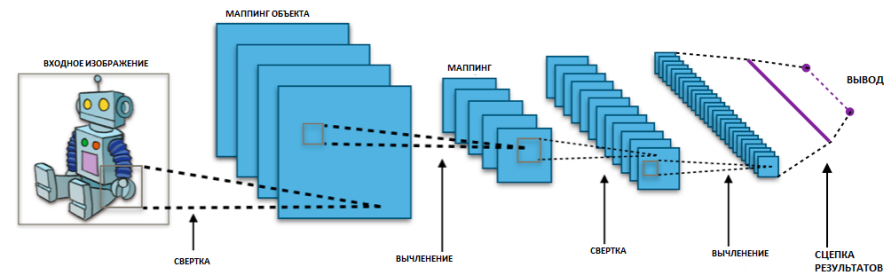
Подходы на основе Deep Learning и Convolutional Neural Network

Deep Learning – это совокупность методов машинного обучения (с учителем, с частичным привлечением учителя, без учителя, с подкреплением), основанных на обучении представлением, а не специализированным алгоритмам под конкретные задачи.

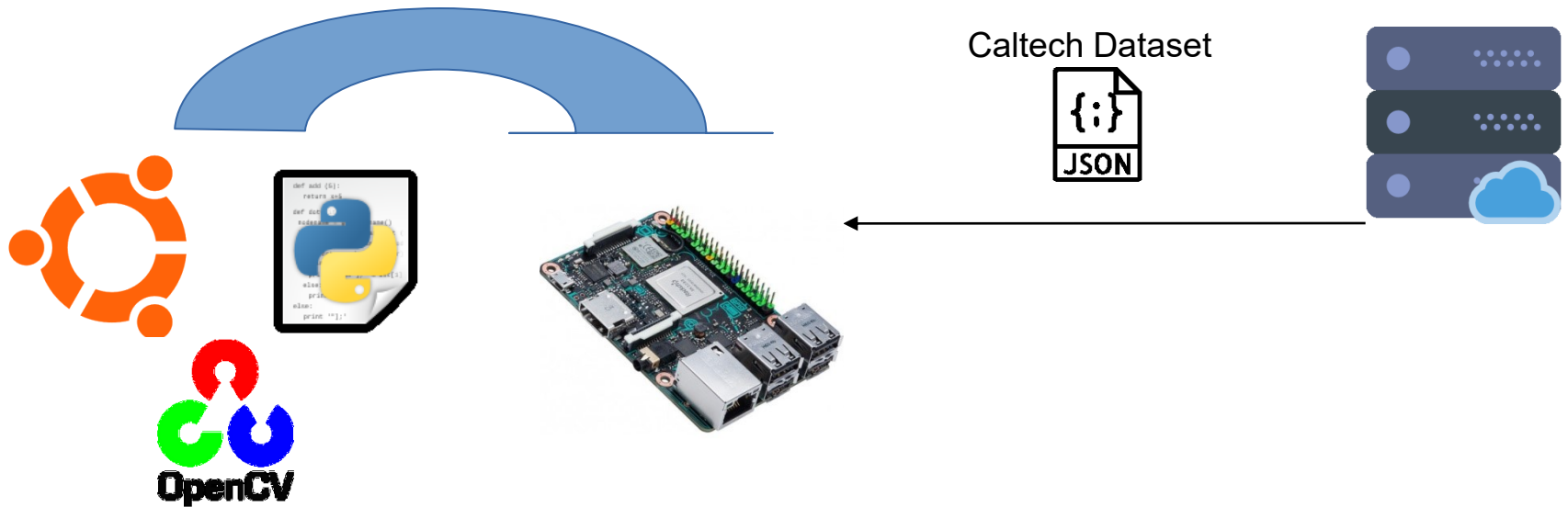


С помощью комплексных методов глубокого обучения можно научить систему определять правильное числовое представление объекта.

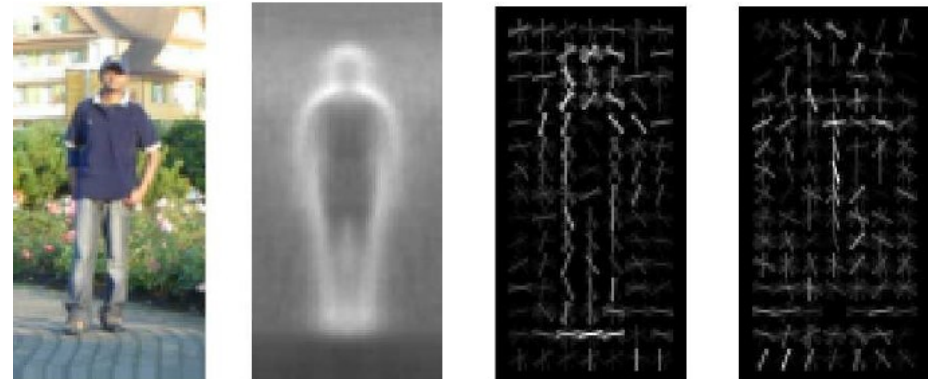
Сверточная сеть является развитием Глубинных сетей имеющая операцию свёртки, которая каждый фрагмент изображения умножает на матрицу свёртки поэлементно, результат суммируется и записывается в аналогичную позицию выходного изображения.



Обнаружение пешеходов используя OpenCV + HOG + SVM + Caltech Dataset



Дескриптор HOG является описанием визуальных особенностей содержимого изображений, он описывает элементарные характеристики, такие как форма, цвет, текстура или движение. Дескрипторы HOG характеризуют сцену, показанную на изображении, например пешехода, идущего по улице.

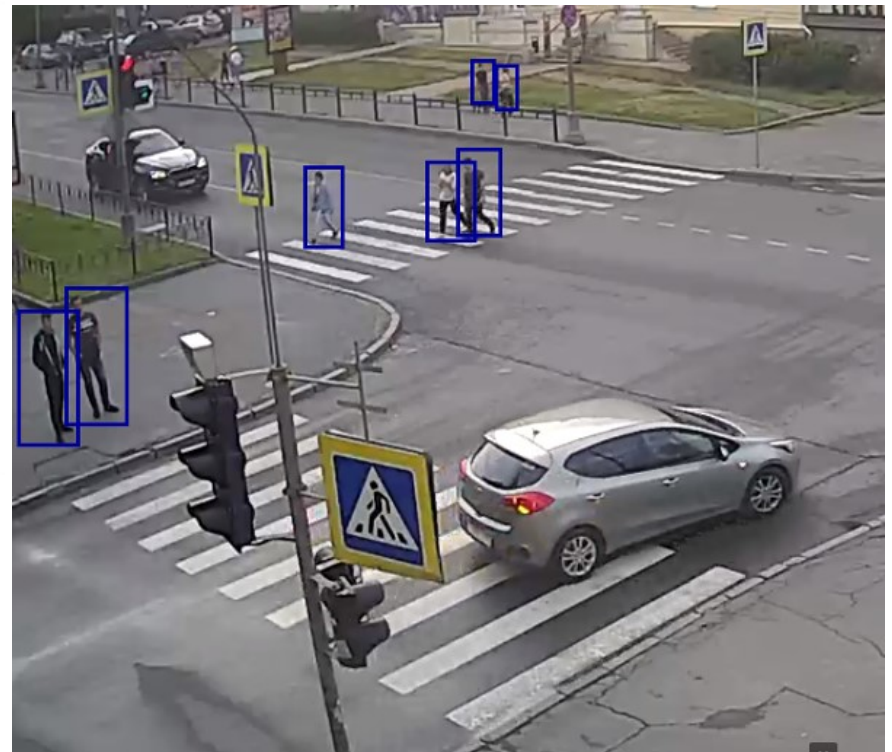
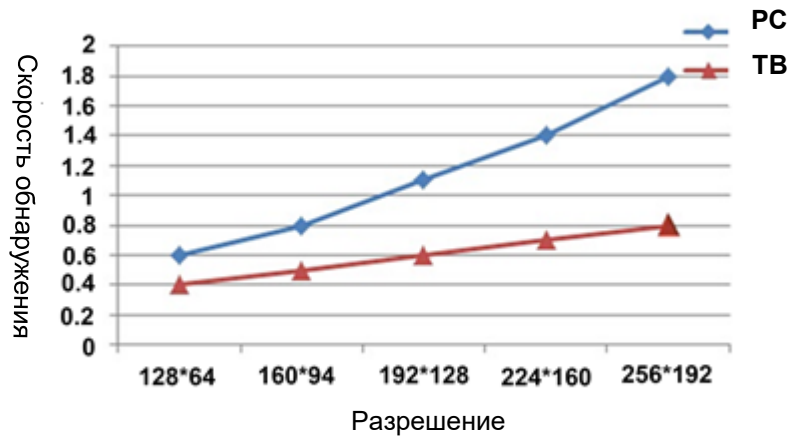


Исходное изображение Градиентное представление Взвешенный позитивный класс Взвешенный негативный класс

Результаты тестов по обнаружению пешеходов

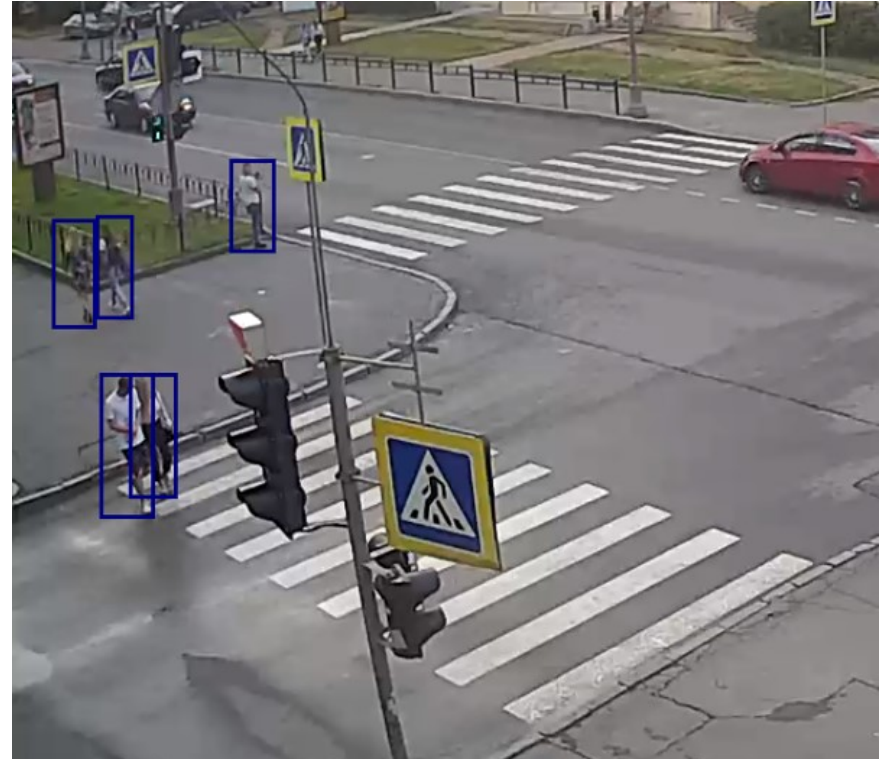
■ Применение HOG + SVM в потоковом видео с использованием OpenCV 3.4 с использованием набора данных Caltech Dataset на тестовом оборудовании:

- стационарная машина: ~10 к/сек
- Tinker Board обнаруживает со скоростью ≤ 1 к/сек



Обнаружение пешеходов с помощью TensorFlow + Detection Model Zoo + COCO Dataset

- Tensorflow это библиотека с открытым исходным кодом, имеющее собственное API и поддерживающий библиотеку OpenCV.
- Tensorflow Detection Model Zoo представляет собой коллекцию пред-обученных моделей совместимых с Tensorflow Object Detection API.
- `ssd_mobilenet_v2_coco` `faster_rcnn_inception_v2` являются наиболее оптимизированными моделями для обнаружения объектов на мобильных устройствах и микрокомпьютерах

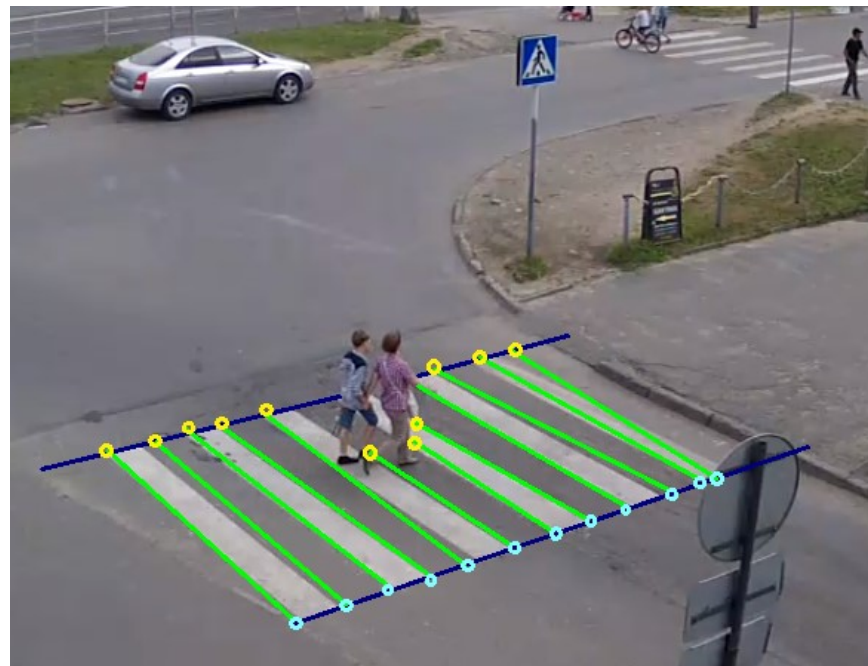


Используя данные модели TinkerBoard обнаруживает объекты со скоростью 4-5 к/сек

Обнаружение пешеходного перехода

Обучение собственной модели:

- Выборка 1000-2000 изображений пешеходной «зебры» со статического изометрического плана
- Распределение набора данных по позитивным и негативным результатам
- Разметка данных с помощью инструментов Tensorflow
- Обучение собственной модели на основе существующих в Tensorflow Model Zoo
- Запуск программы по обнаружению



Вычисление позы пешехода

Наиболее общая поза пешехода по которой система может «догадаться» о намерении перейти проезжую часть является поза ожидания - «стоячая» поза

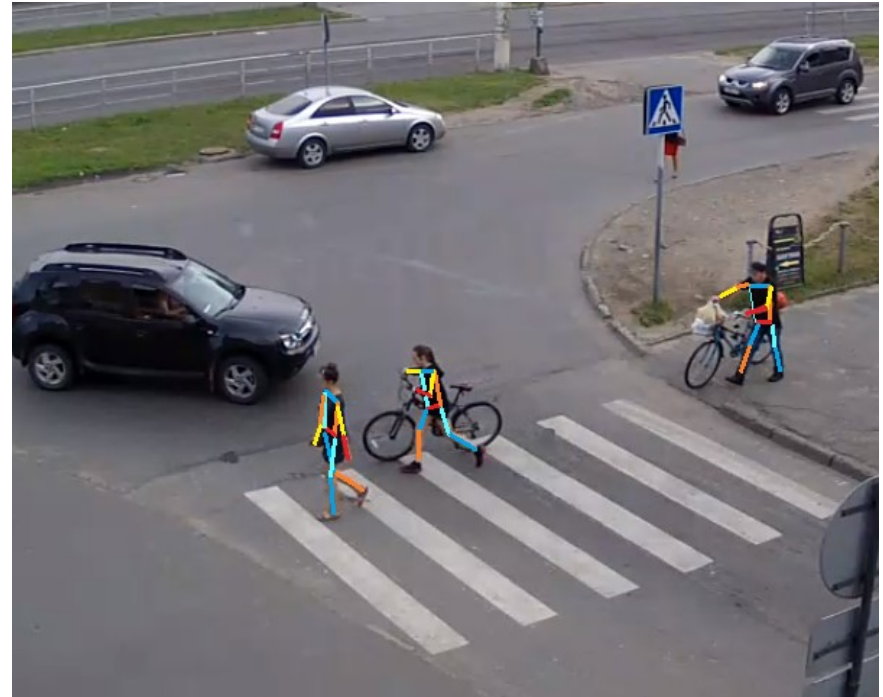
Для оценки человеческой позы используется библиотека Tensorflow - Pose Estimation (tf-pose-estimation), содержащая ряд обученных моделей, основанных на модели OpenPose.

Модели не распознают объект на изображении, а оценивают расположение суставов, на основе этих данных делается вывод какую позу принял объект.

Одна из моделей Pose Estimation основана на mobilenet и оптимизированная для работы на маломощных устройствах

Скорость работы на TinkerBoard составляет ~4 к/сек

Ваш логотип



Объединение модулей в единую систему с помощью Tensorflow + YOLO + Keras

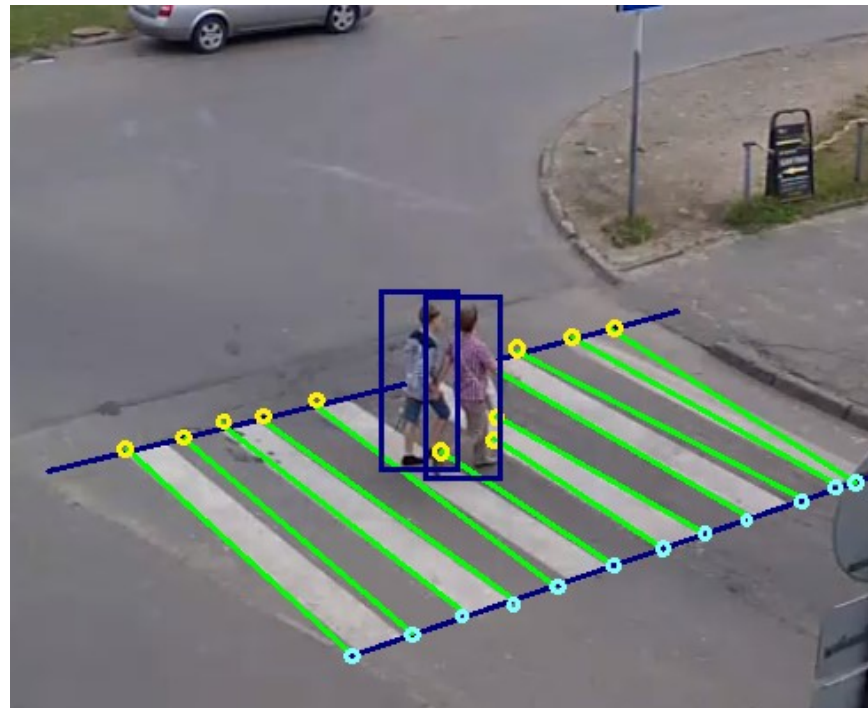
Keras – открытая нейросетевая библиотека, представляющая собой надстройку над различными фреймворками (Tensorflow, Teano)

YOLO — система по обнаружению объектов (TinyYOLO вариант для микрокомпьютеров Raspberry Pi, Tinker Board и пр.)

Обучение новой модели:

- загрузка модели DarkNet и весов
- генерация файла весов в конфигурационный файл для использования в Keras
- размещение изображения которое необходимо обнаружить и запуск инструмента yolo.py для обучения

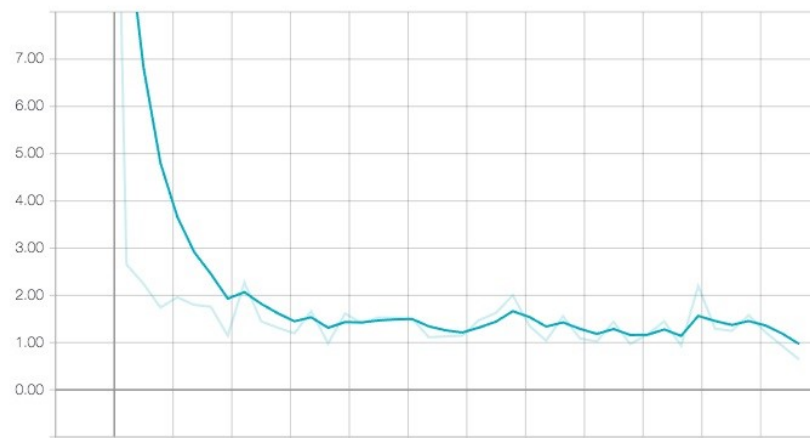
Скорость обнаружения объектов на TinkerBoard составляет ~1.5-2 к/сек



Анализ результатов и вывод

Узкие места системы

- Производительность железа
- Графический ускоритель
- Уровень освещенности
- Блики
- Набор данных для обучения



Задачи необходимые решить для усовершенствования и эксплуатации модели

- Определение ТОЛЬКО тех пешеходов которые находятся рядом с пешеходным переходом
- Подсчет количества пешеходов до пересечения проезжей части и после
- Более точное определение позы пешеходов
- Отслеживание перемещения пешеходов и их позы
- Автоматическая регулировка освещения в темное время суток

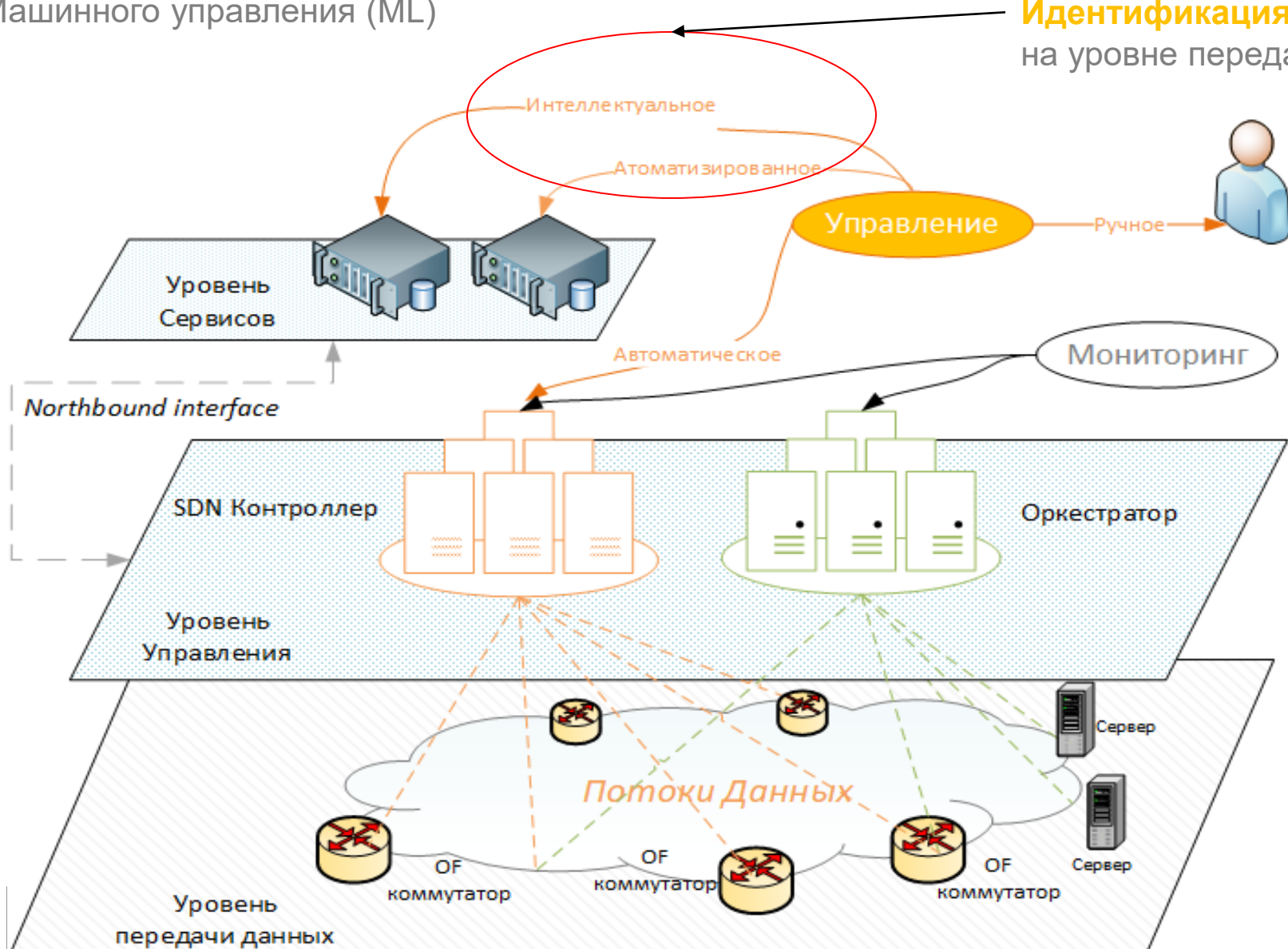
Анализ проблем современных телекоммуникационных сетей



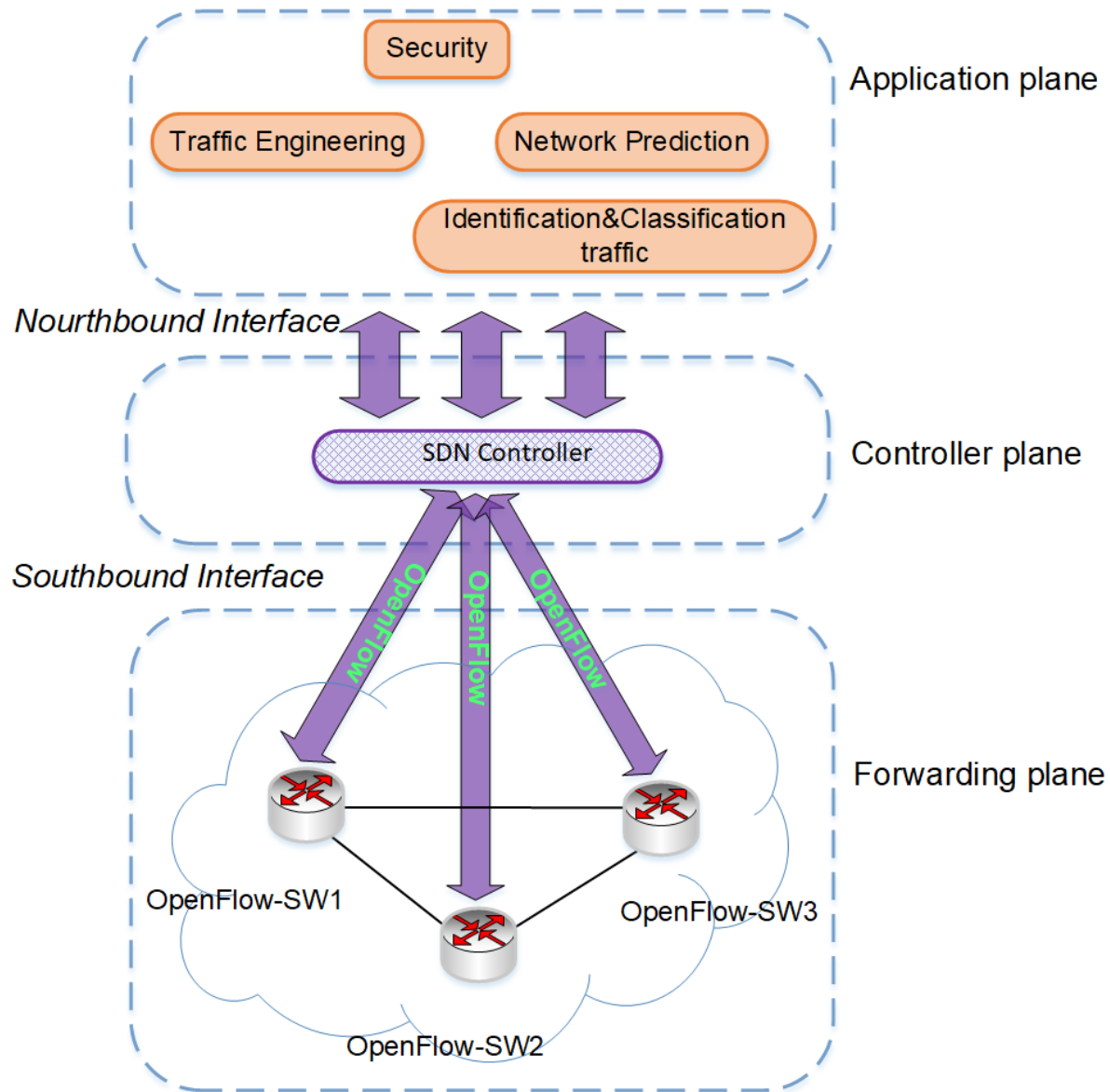
Управление и мониторинг в сетях SDN/NFV

Интеллектуальное управление – основанное на алгоритмах Машинного управления (ML)

Требуется **однозначная Идентификация** потоков на уровне передачи данных



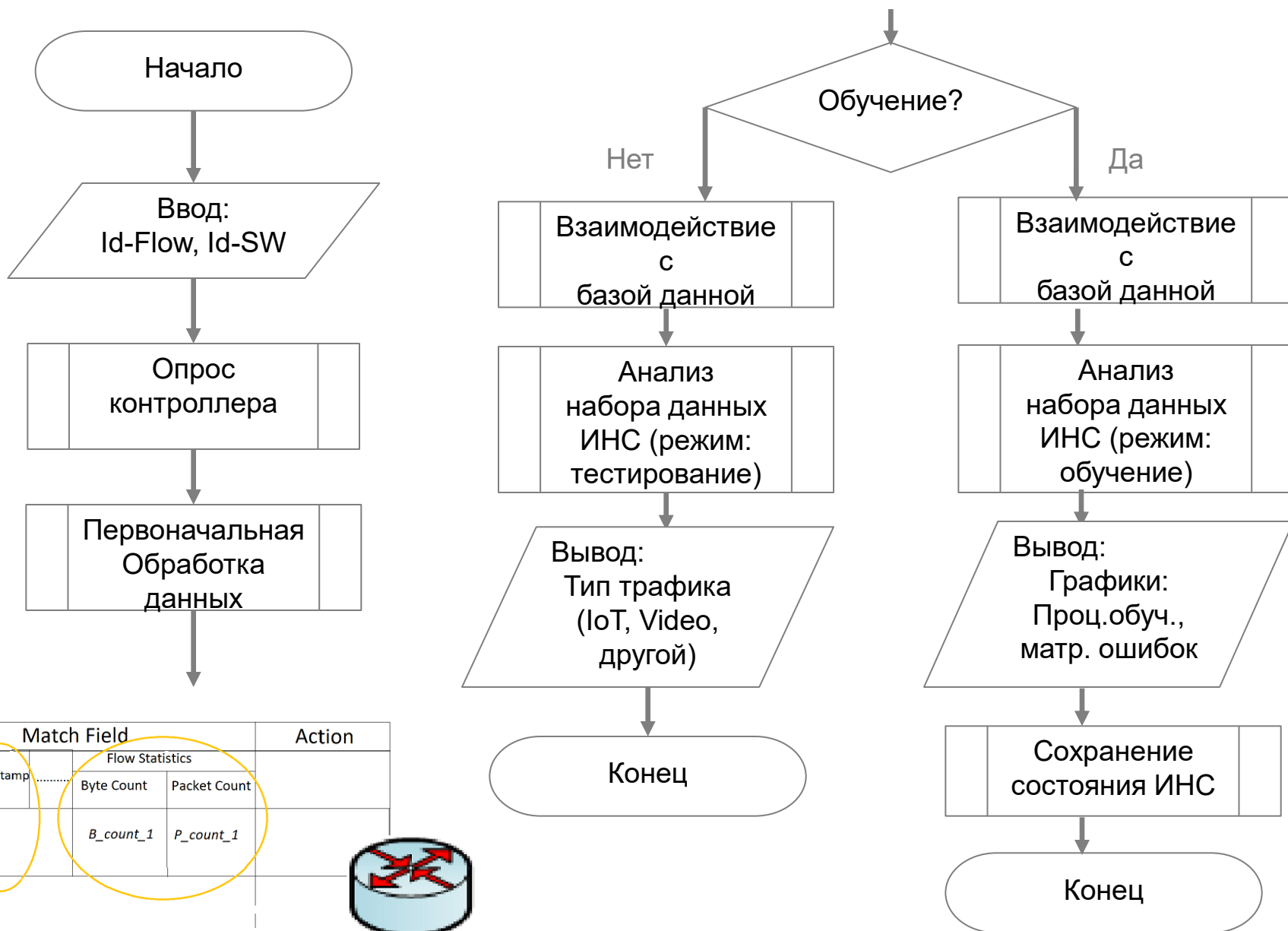
Концепция «Machine Learning+SDN»



Методы машинного обучения для прогнозирования развития сети

Обучение с учителем	Обучение без учителя	Обучение с подкреплением	Обучение с частичным привлечением учителя
Метод опорных векторов	Подход кластеризации по методу К-средних	Метод «Q-обучение»	Метод рекуррентных нейронных сетей
Метод деревьев решений	Метод самоорганизующейся карты Кохонена	Метод глубокого обучения с подкреплением	
Метод ансамбля	Метод скрытой марковской модели		
Метод глубокого обучения с учителем	Метод ограниченной модели Больцмана		Метод многопараметрического анализа
Метод сверточных нейронных сетей	Метод глубокого обучения без учителя		
	Метод составного автоэнкодера		
	Метод сети глубокого доверия		

Разработка метода идентификации потоков ИВ



#	Match Field				Action
	In Port	TimeStamp	Flow Statistics		
			Byte Count	Packet Count	
1			B_count_1	P_count_1	



Разработка метода идентификации потоков ИВ (Математический аппарат первоначальной обработки данных)

Пусть, $PacketCount_delta - PC_{delta}$,

$ByteCount_delta - BC_{delta}$, а

$TimeStamp_delta - TS = 1 [sec.] = const$, тогда:

	[TimeStamp]	[ByteCount]	[PacketCount]	
$DataSet_{RQ} =$	$TimeStamp_{11}$	$ByteCount_{12}$	$PacketCount_{13}$	
	$TimeStamp_{21}$	$ByteCount_{22}$	$PacketCount_{23}$	
	
	$TimeStamp_{N1}$	$ByteCount_{N2}$	$PacketCount_{N3}$	
→	$\begin{cases} BC_delta_{N2} = ByteCount_{N2} - ByteCount_{(N-1)2}, & \text{if } \uparrow \\ PC_delta_{N2} = PacketCount_{N2} - PacketCount_{(N-1)2}, & \text{if } \uparrow \end{cases}$			
$DataSet_{ML} =$	TS	$BC_{delta12}$	$PC_{delta13}$	
	TS	$BC_{delta22}$	$PC_{delta23}$	
	
	TS	$BC_{delta_{N2}}$	$PC_{delta_{N3}}$	

Архитектура Искусственной Нейронной Сети

	[TypeOfTraffic]	[TimeStamp]	[ByteCount]	[PacketCount]
	<i>IoT</i>	<i>TS</i>	<i>BC_{delta₁₂}</i>	<i>PC_{delta₁₃}</i>
	<i>IoT</i>	<i>TS</i>	<i>BC_{delta₂₂}</i>	<i>PC_{delta₂₃}</i>
	<i>...</i>	<i>...</i>	<i>...</i>	<i>...</i>
	<i>Video</i>	<i>TS</i>	<i>BC_{delta_{N2}}</i>	<i>PC_{delta_{N3}}</i>
	<i>others</i>	<i>TS</i>	<i>BC_{delta_{(N+1)2}}</i>	<i>PC_{delta_{(N+1)3}}</i>

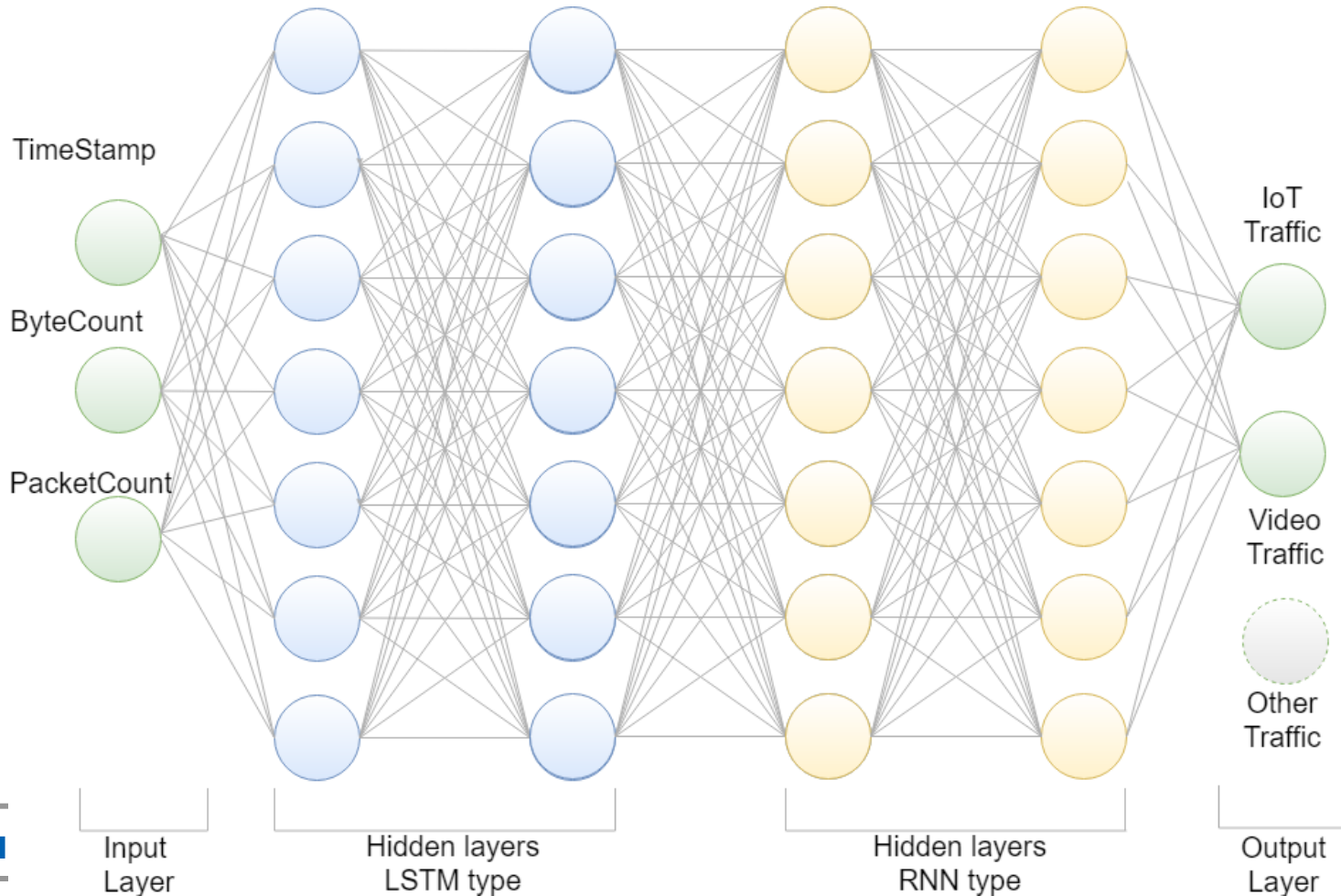
Параметры ИНС:

Оптимизатор: Adam;

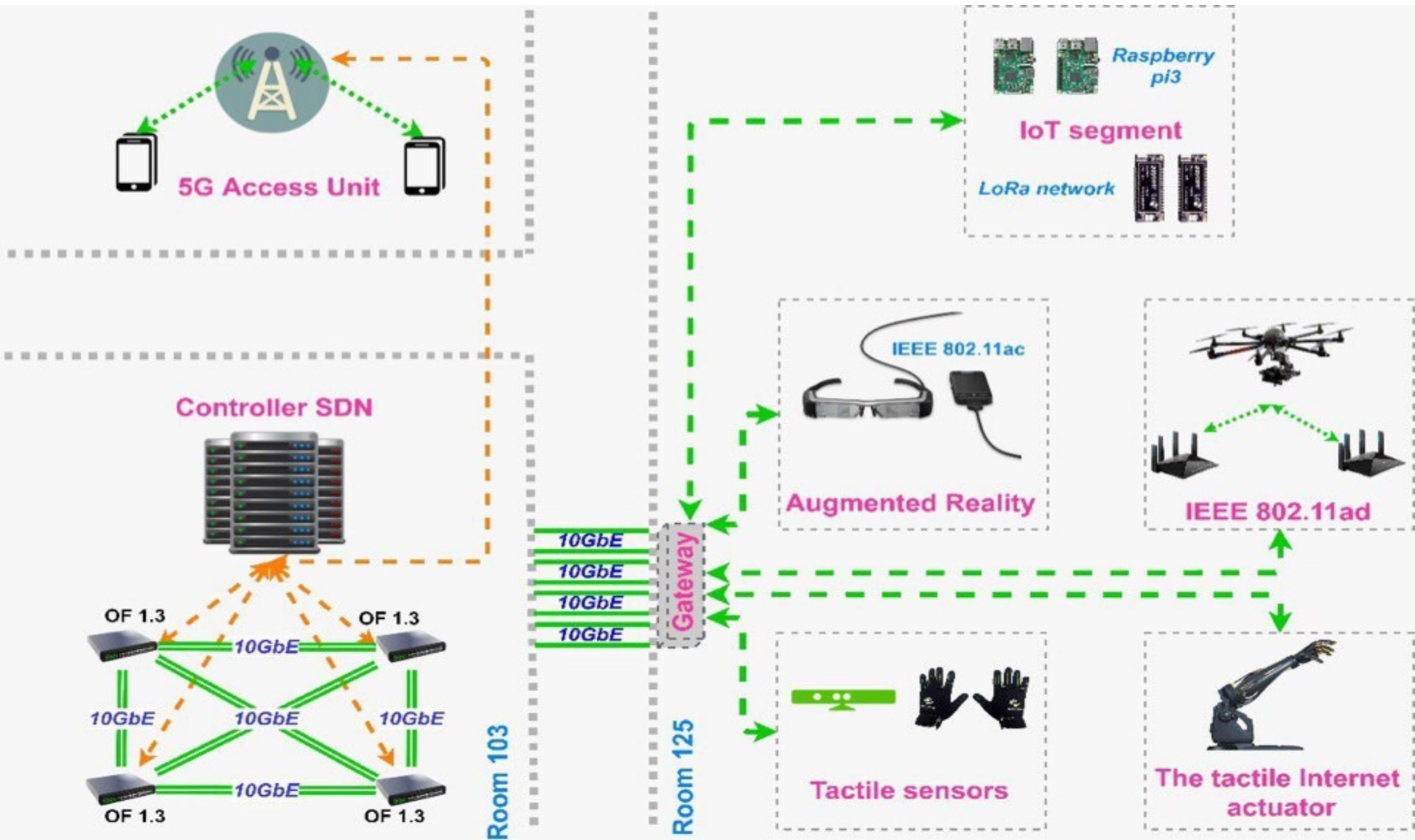
Количество эпох обучения: 40;

Размер «Batch»: 1024

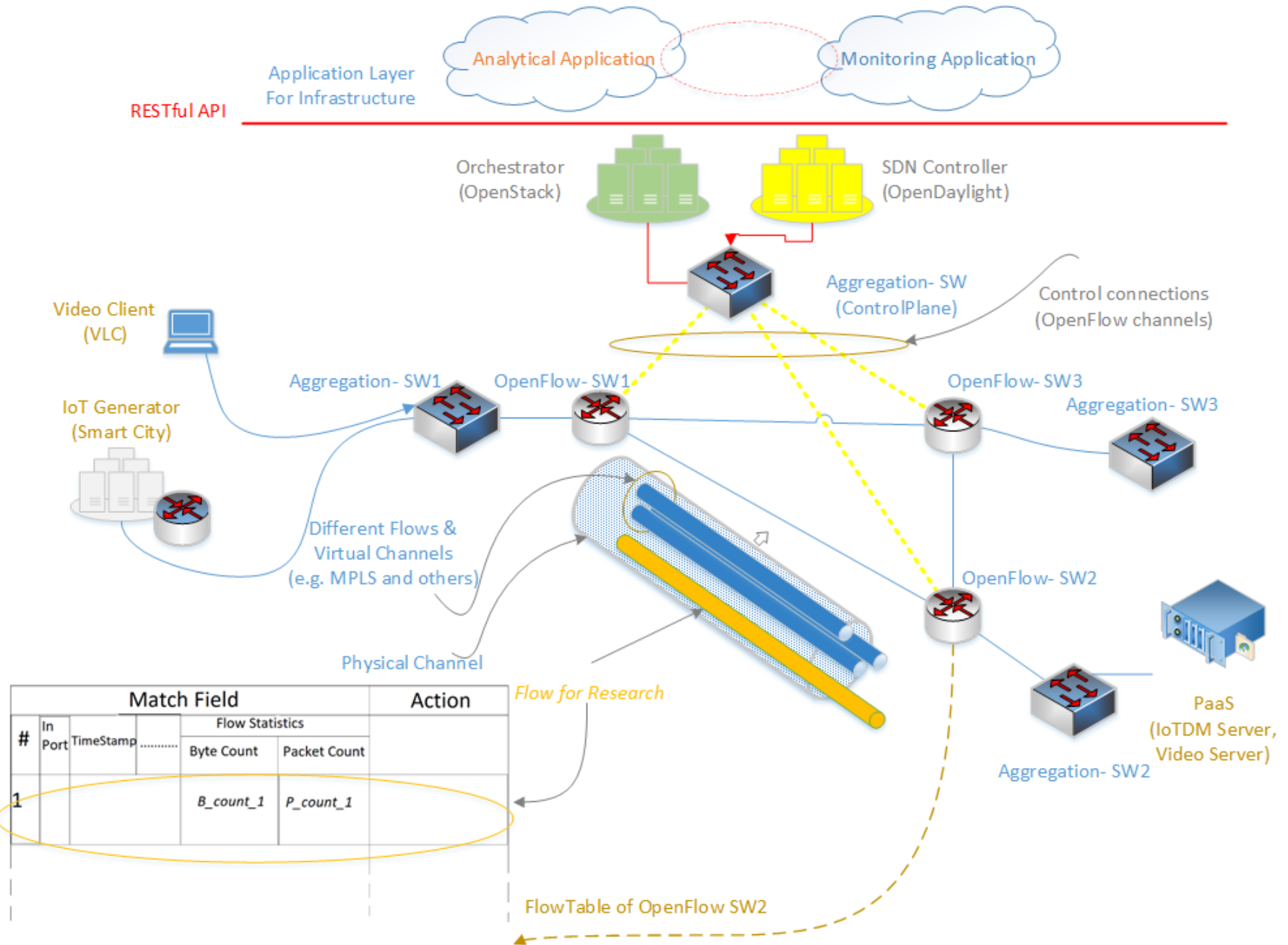
$DataSet_{ML} =$



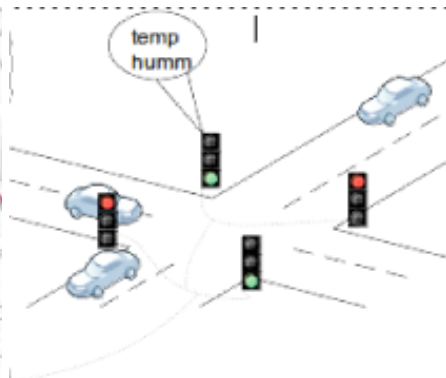
Структура лаборатории РУДН



Модельная сеть



Лабораторный стенд. Трафик генератор ИВ.



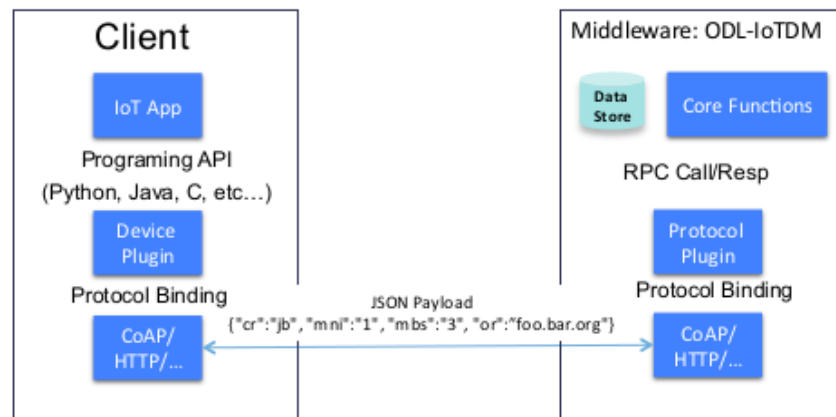
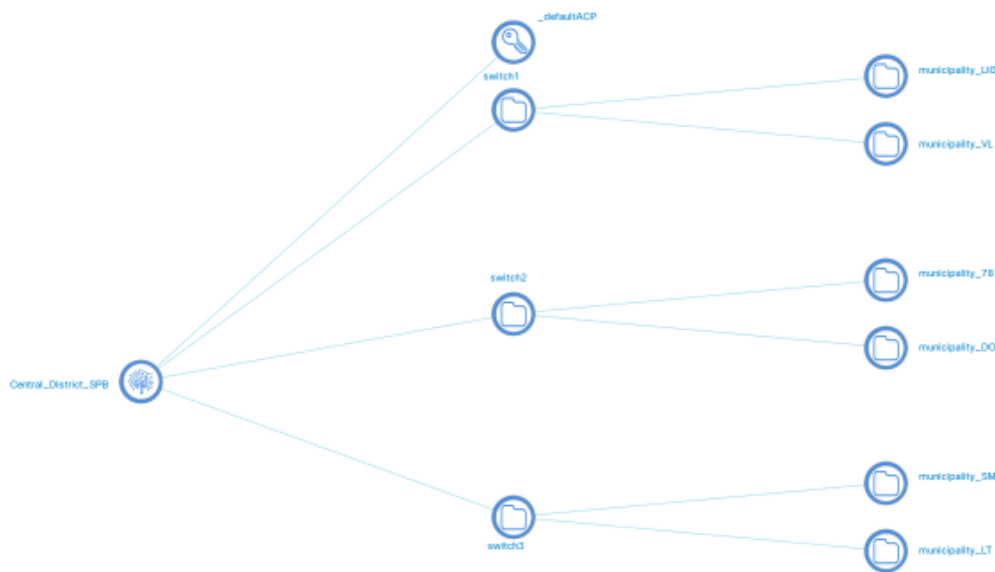
Каждая ИВ имеет три типа датчиков

- ✓ Влажность
- ✓ Температура
- ✓ GPS

Сервер ИВ:

- ✓IoTDM – Internet of Things Data management
- ✓Версия ODL IoTDM – Boron SR2
- ✓Версия OS: Linux Ubuntu 16.0 LTS
- ✓CPU: Intel Xeon(R) E3-1220V2
- ✓RAM: 14 Гб.

Схема взаимодействия «Вещь – IoTDM»



Параметры эксперимента

Генерация трафика ИВ

Количество устройств ИВ	Время теста, (сек)	Протокол уровня приложений	Генерация данных от устройств ИВ, (сек)
260	70 000	HTTP 2.0	$\Delta t = 1$

Генерация видео трафика

Количество клиентов	Время теста, (сек)	Протокол уровня приложений	Используемый Видео кодек
1	30 000	RTSP	H.264

Технические характеристики главных элементов модельной сети

SDN - Контроллер:

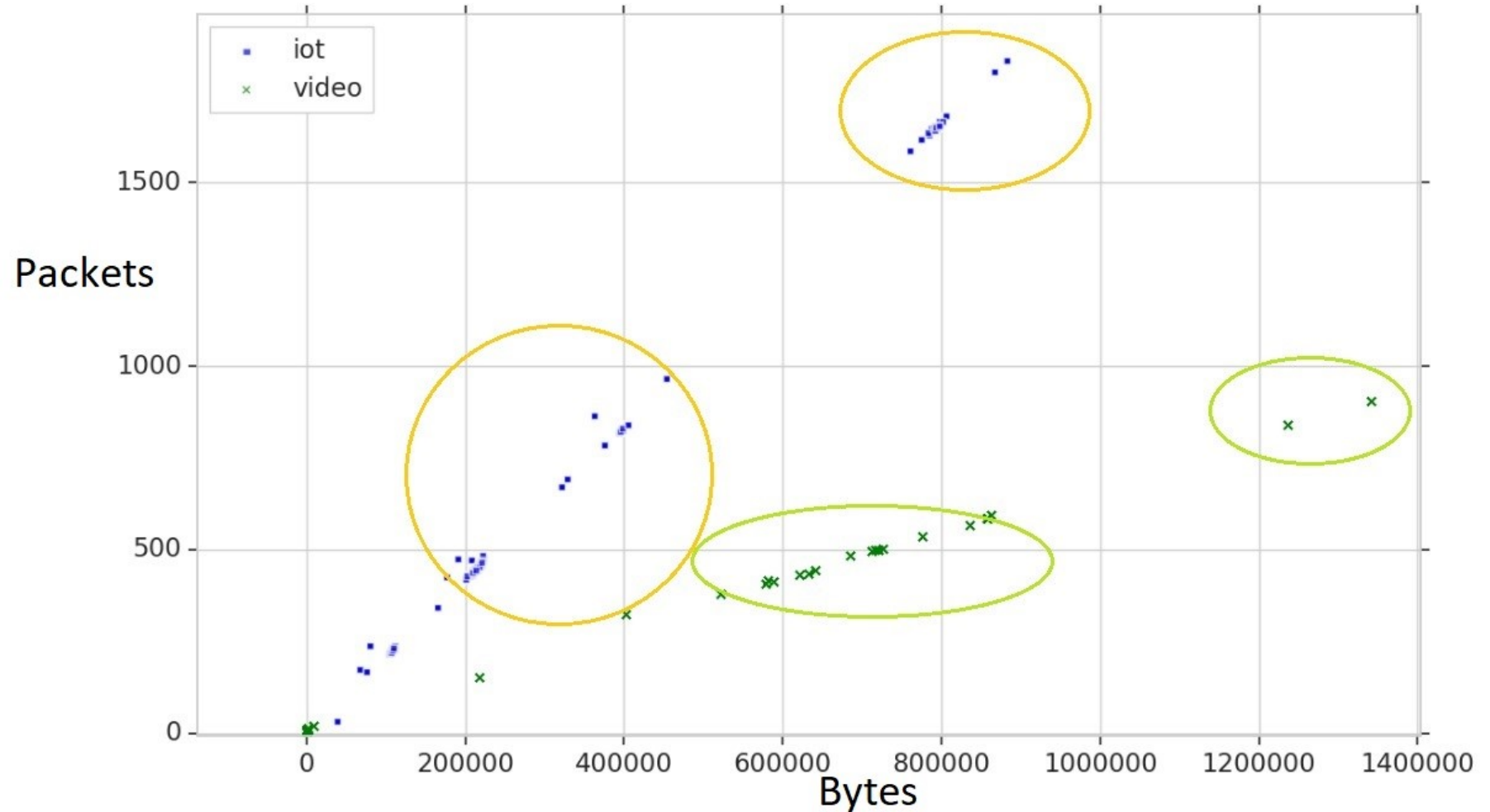
- CPU: Intel Xeon(R) E3-1220V2
- RAM: 16 Гб
- ОС: Linux Ubuntu 14.04 LTS
- Версия: Beryllium SR4
- API: RESTful, Java

Сервер Аналитики:

- CPU: Intel Core i5
- RAM: 8 Гб
- ОС: Linux Ubuntu 16.04 LTS
- Версия: Beryllium SR4
- API: RESTful, Python 2.7

Результат апробации метода (этап: предварительная обработка данных)

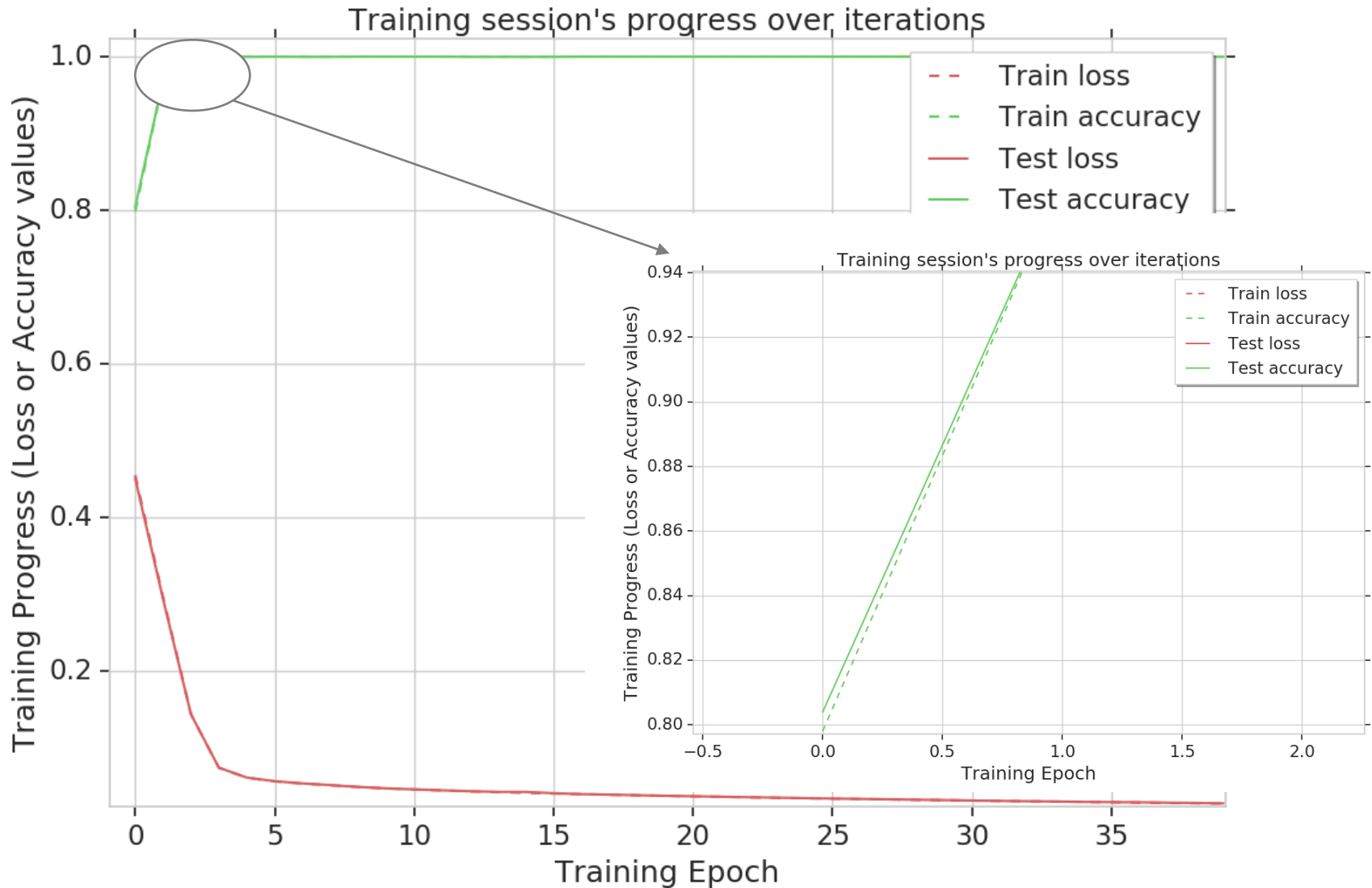
Диаграмма разброса значений $DataSet_{ML}$



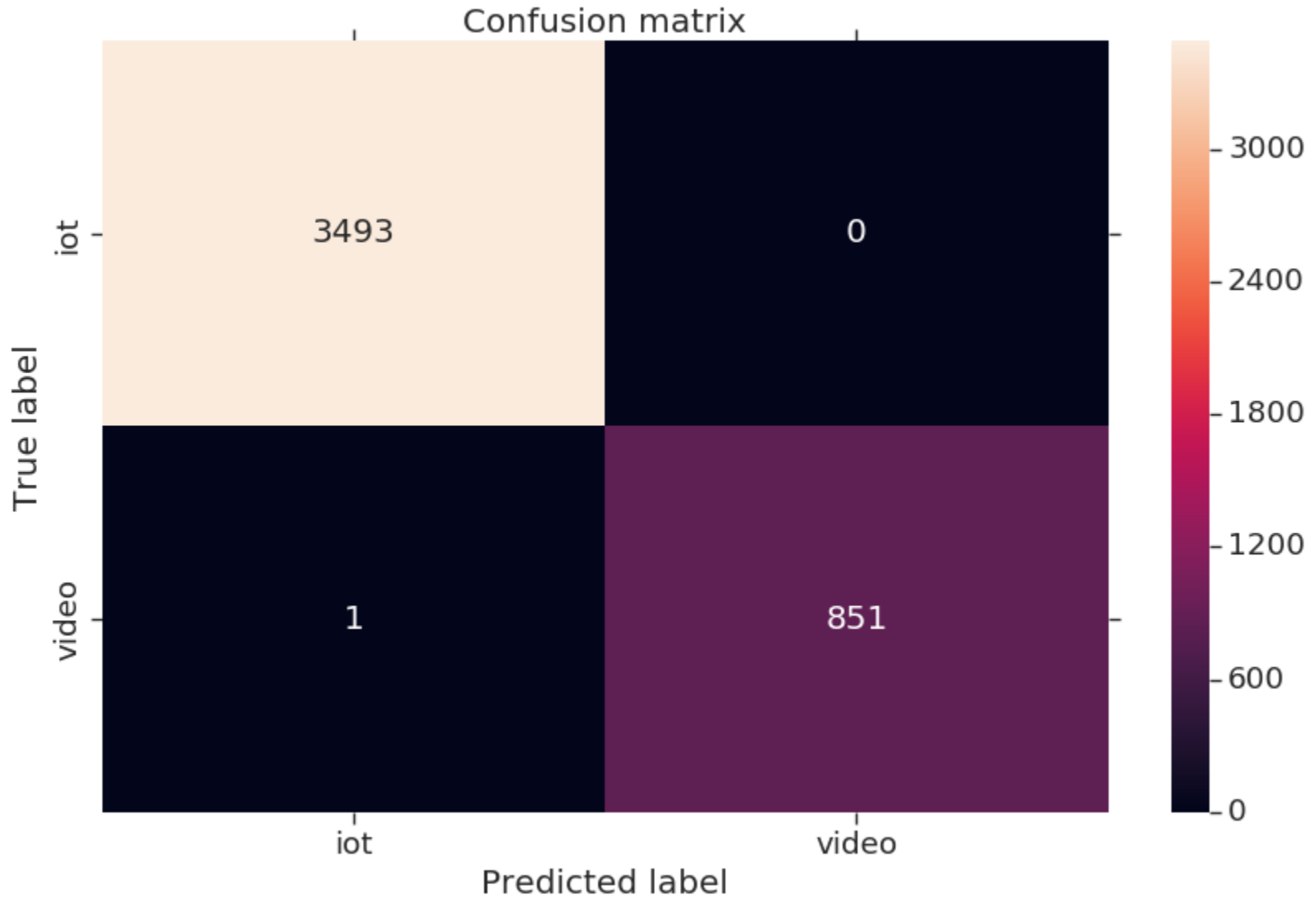
Results

(Learning and testing)

Identification probability : 99,97 %



Results (Testing)



Этапы **НАХОЖДЕНИЯ** корреляции параметров программно-определяемой сети

Этап 1. Формирование многомерной матрицы данных для сравнения

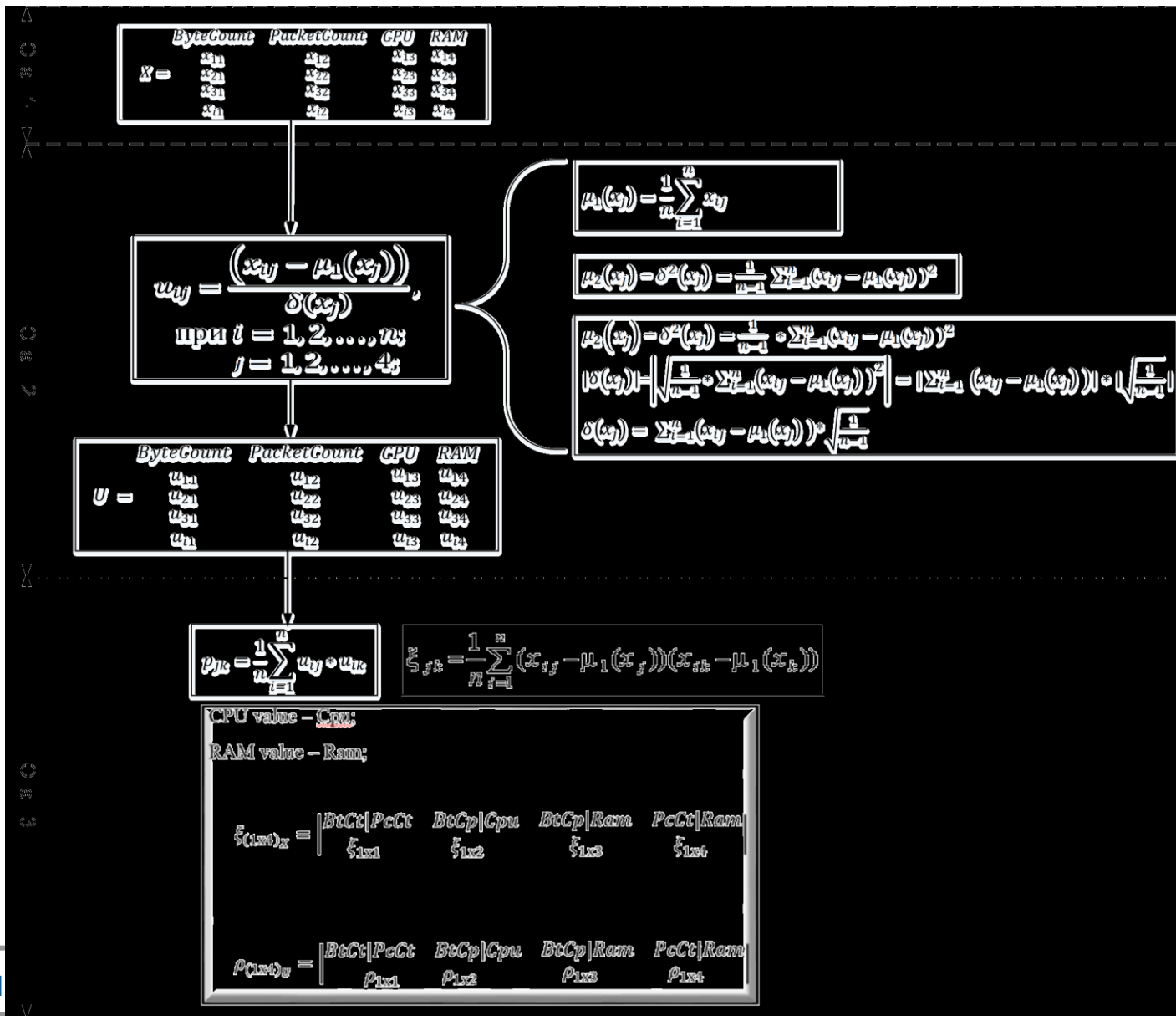


Этап 2. Приведение сформированной матрицы к нормированной



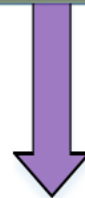
Этап 3. Вычисление корреляционного коэффициента

Математический аппарат корреляции между параметрами SDN



Этапы прогнозирования нагрузки в SDN

Этап 1. Сбор и формирование выборок для обучения и тестирования рекуррентной нейронной сети

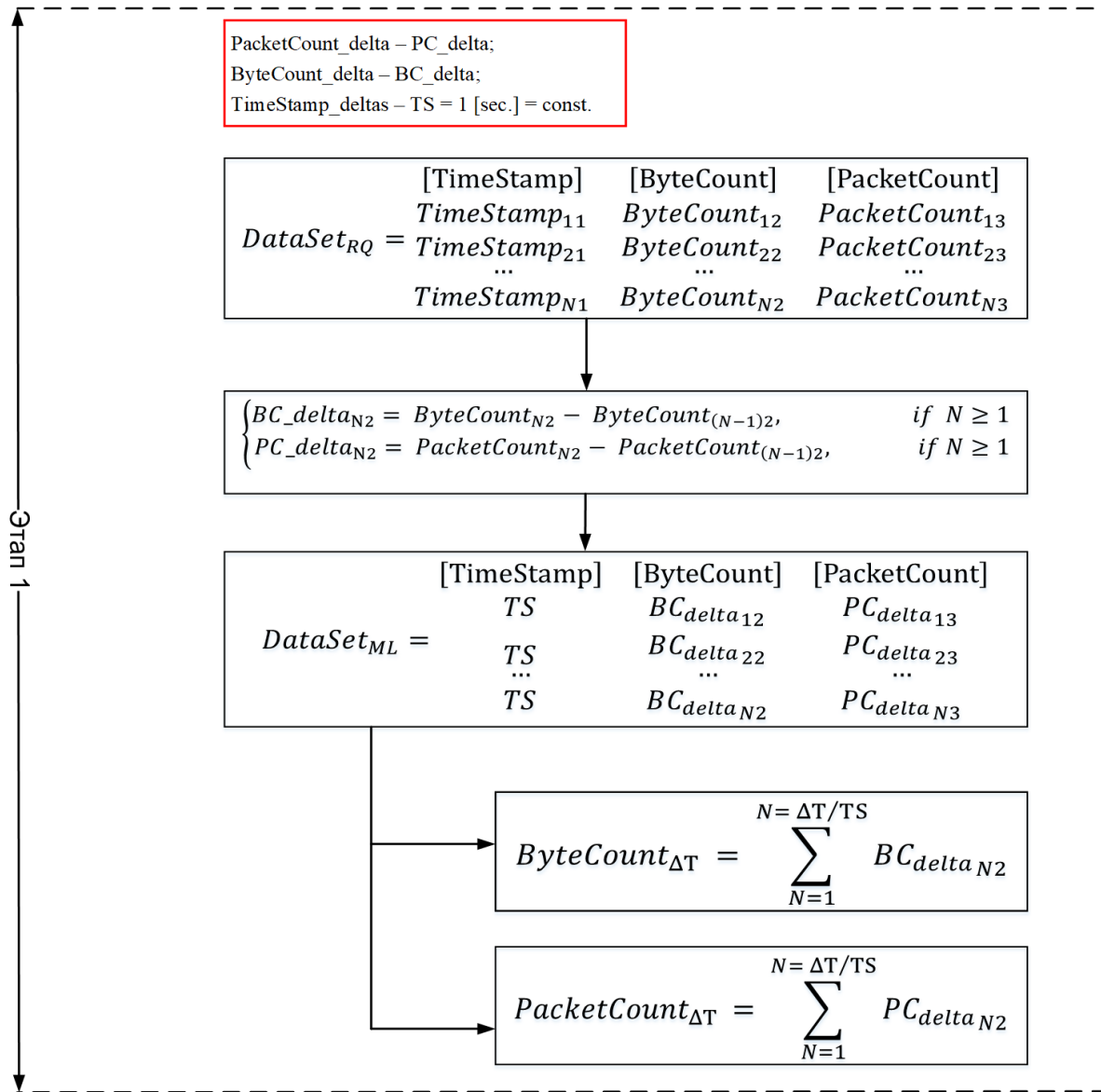


Этап 2. Обучение рекуррентной нейронной сети

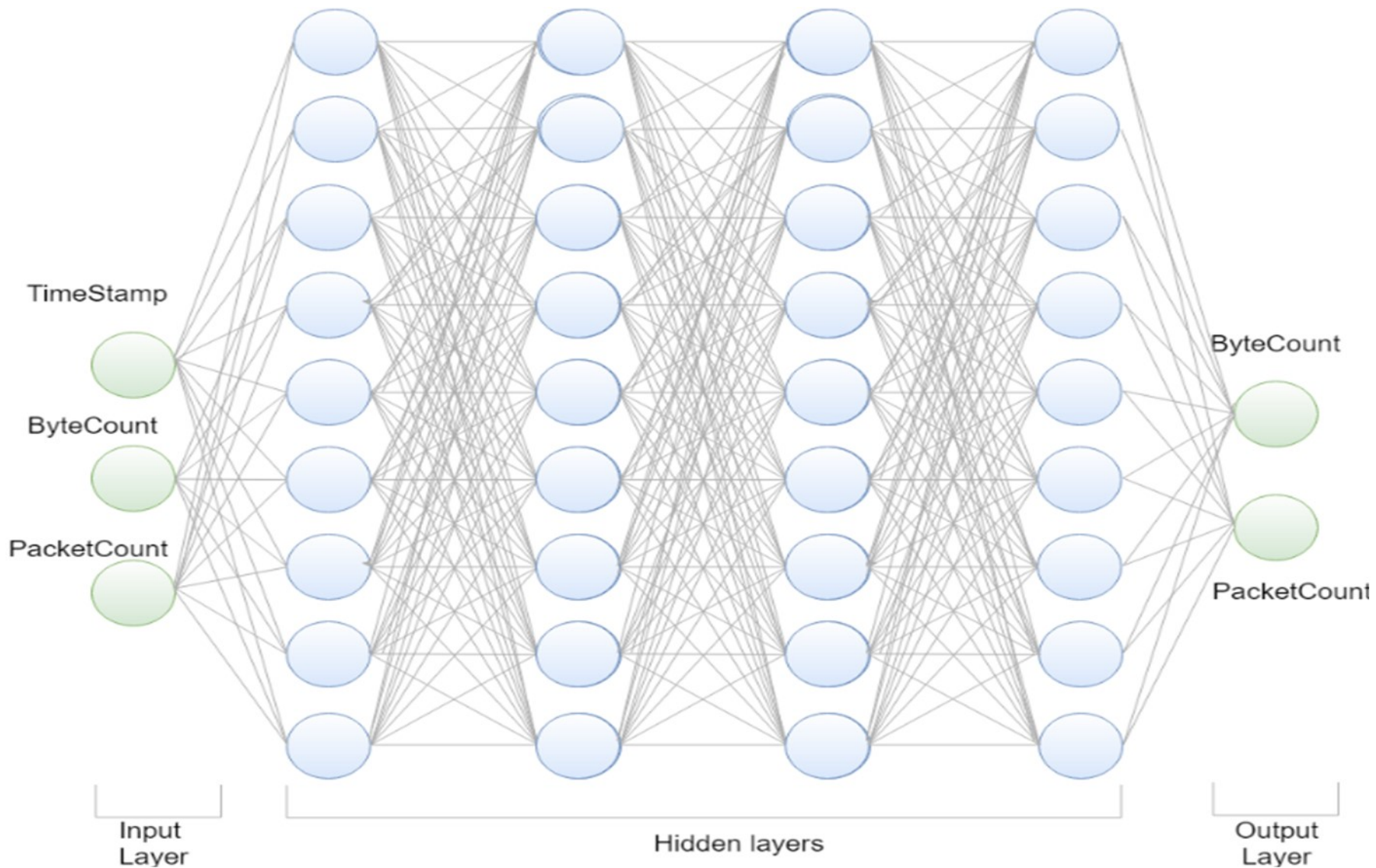


Этап 3. Тестирование рекуррентной нейронной сети

Математический аппарат прогнозирования нагрузки в SDN



Прогнозирование нагрузки в SDN



Натурный эксперимент

Network Parametr:

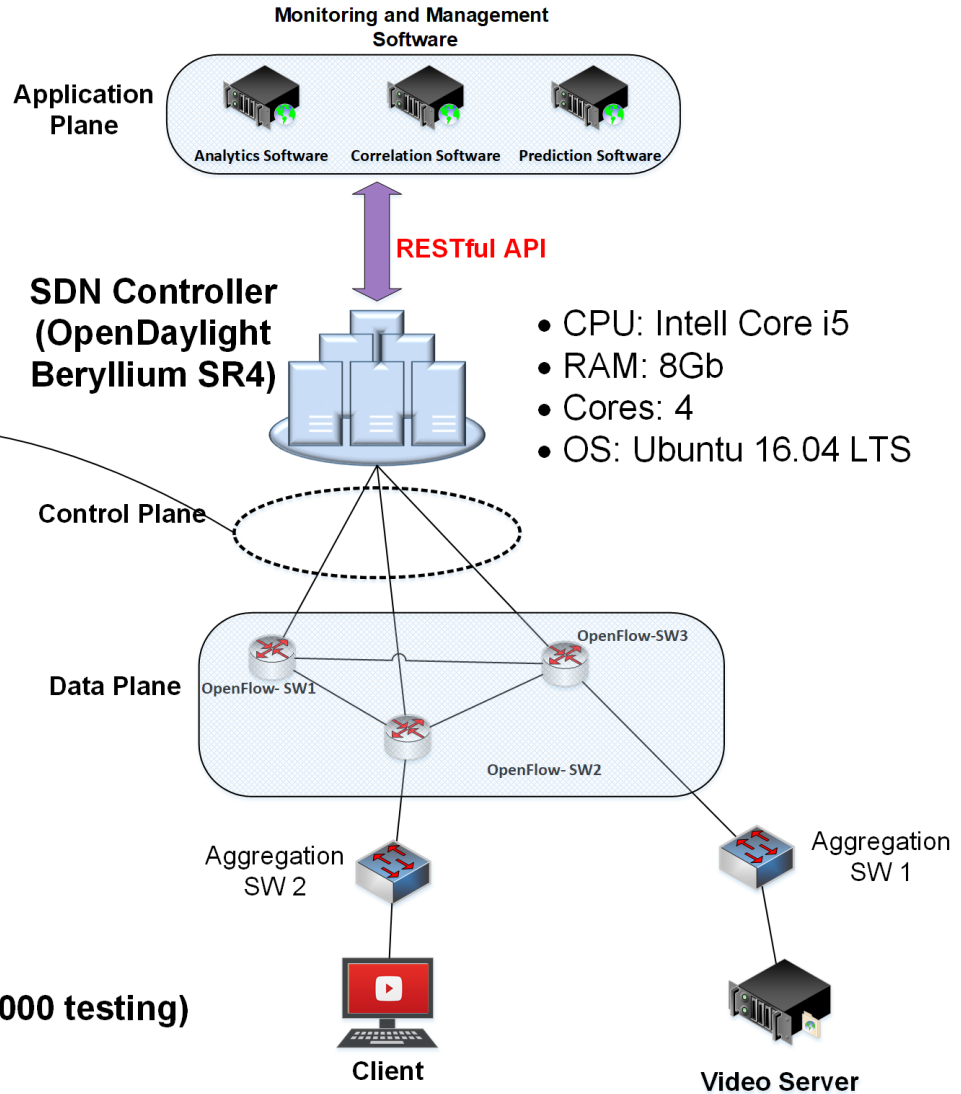
- PacketCount Pc)
- ByteCount (Bt)
- TimeStamp

Hardware Parametr:

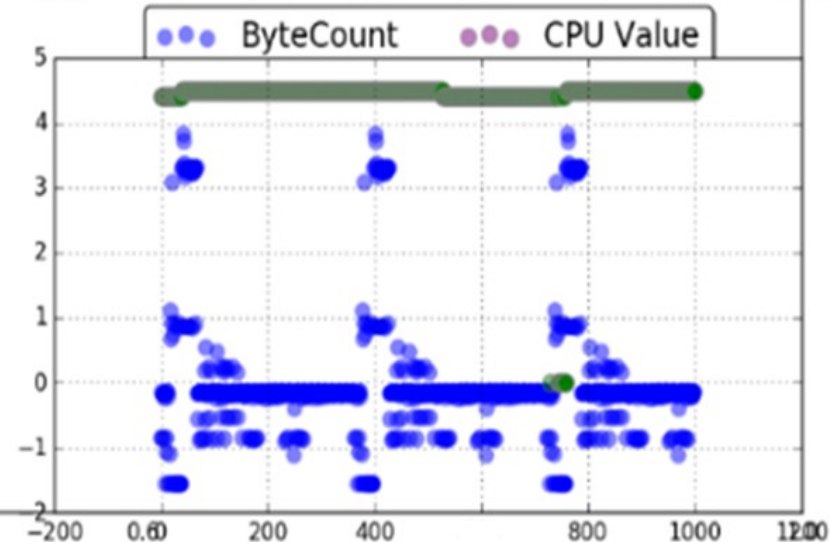
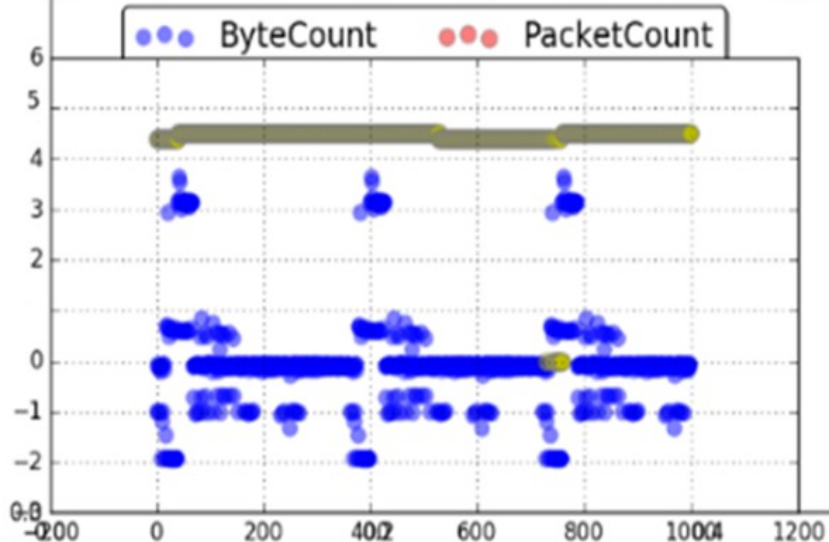
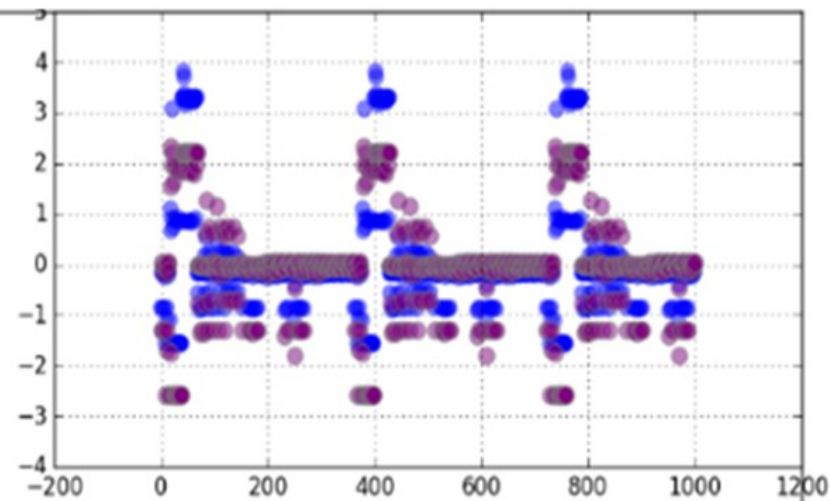
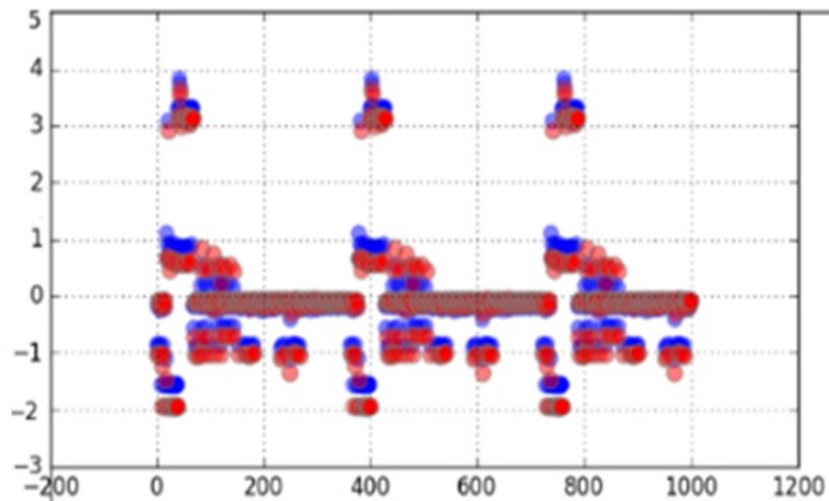
- CPU (%)
- RAM (Gb)

FlowTable of <i>OpenFlow SW1 / OpenFlow SW2 / OpenFlow SW3</i>							
#	In Port	TimeStamp	Match Field		Flow Statistics		Action
			Byte Count	Packet Count	Byte Count	Packet Count	
1		Time_Stamp_1	B_count_1	P_count_1			
Flow for Research							

- T=27 h
- $\Delta T=1$ sec
- V=100 000 value (80 000 training/ 20 000 testing)
- 20 epochs
- 1 epoch = 78 batch
- 1 batch = 256 training/testing value
- $T_{learn} = 0,4$ h
- $T_{prediction} = 5,55$ h

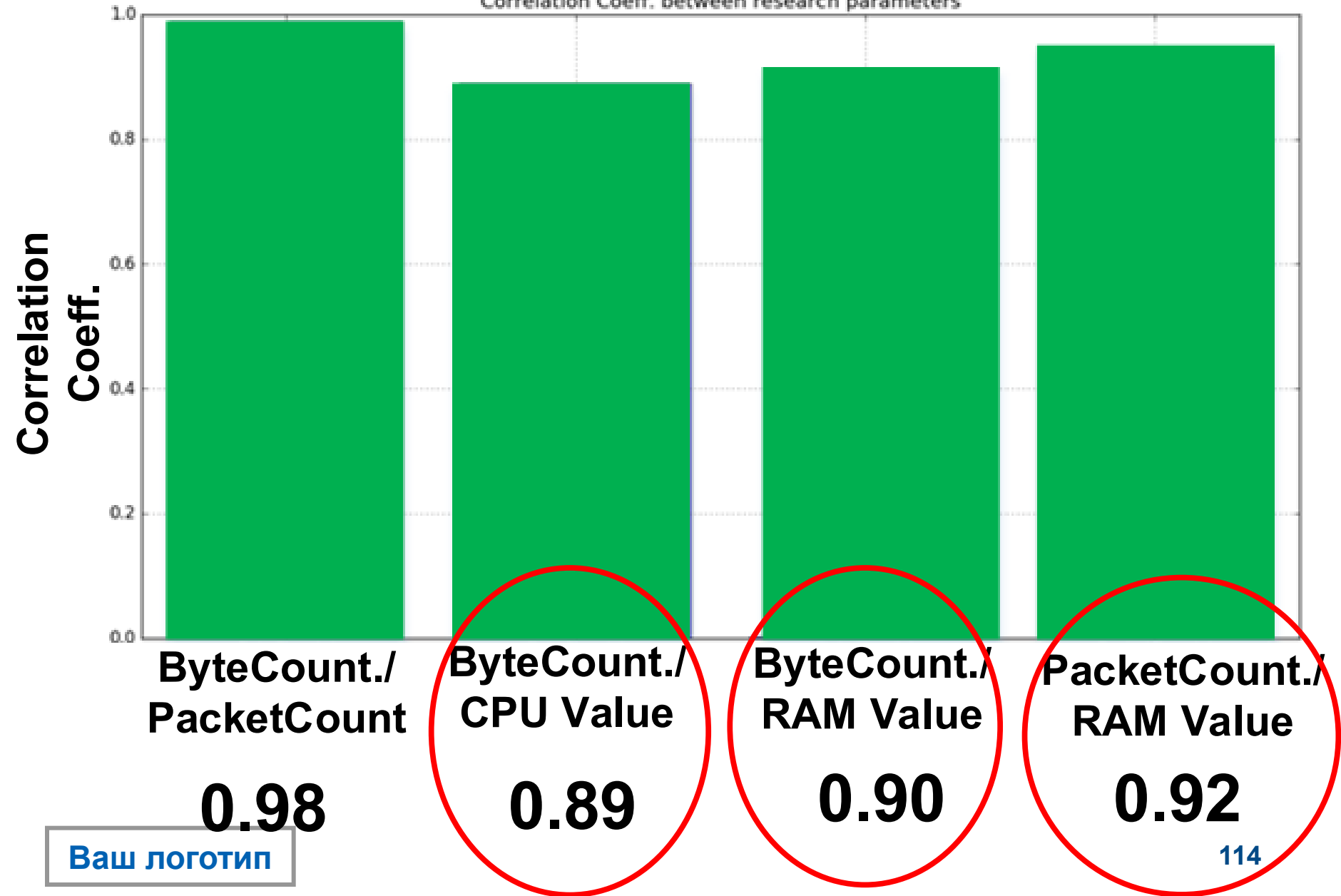


Результаты многопараметрического анализа

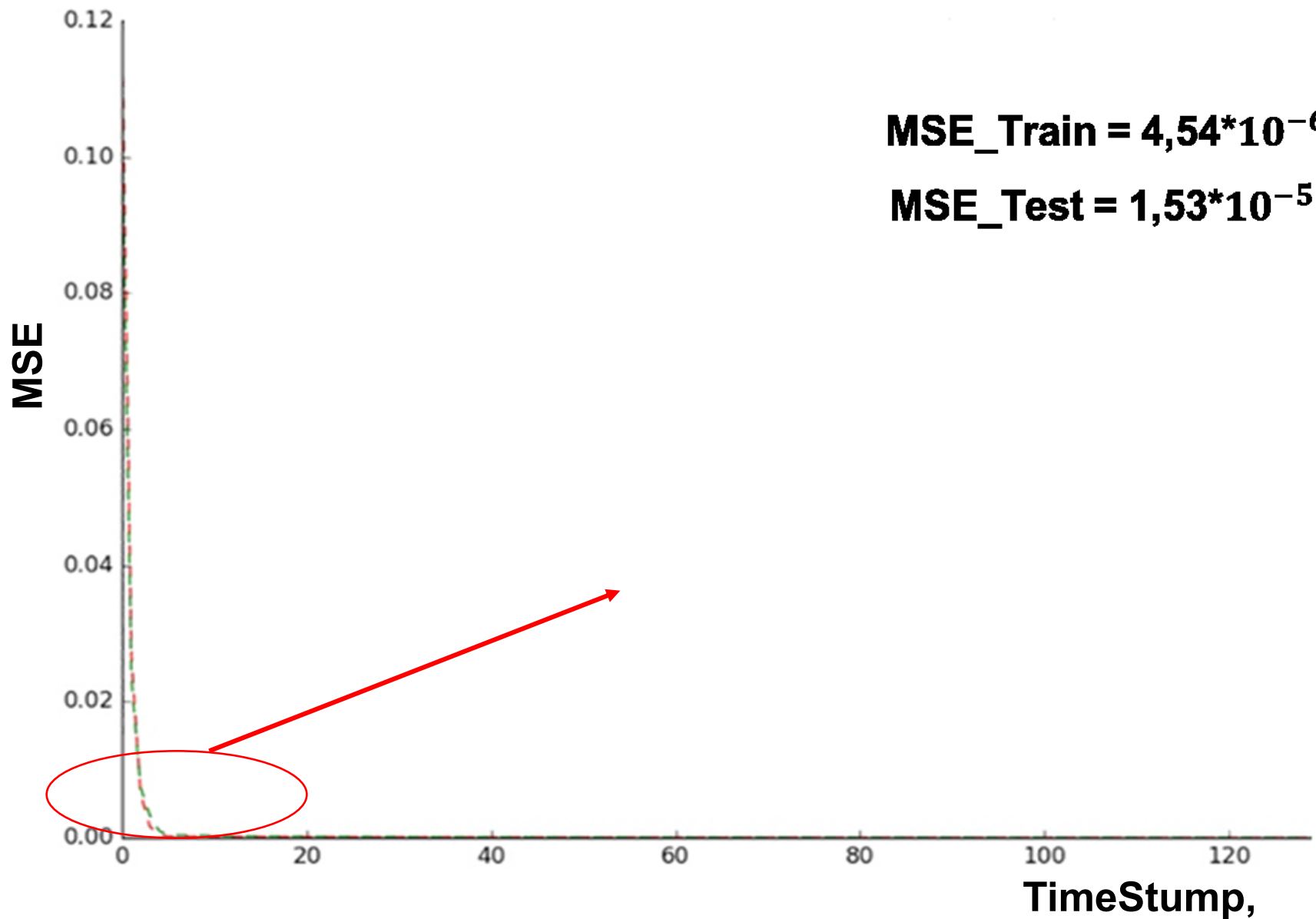


Результаты многопараметрического анализа

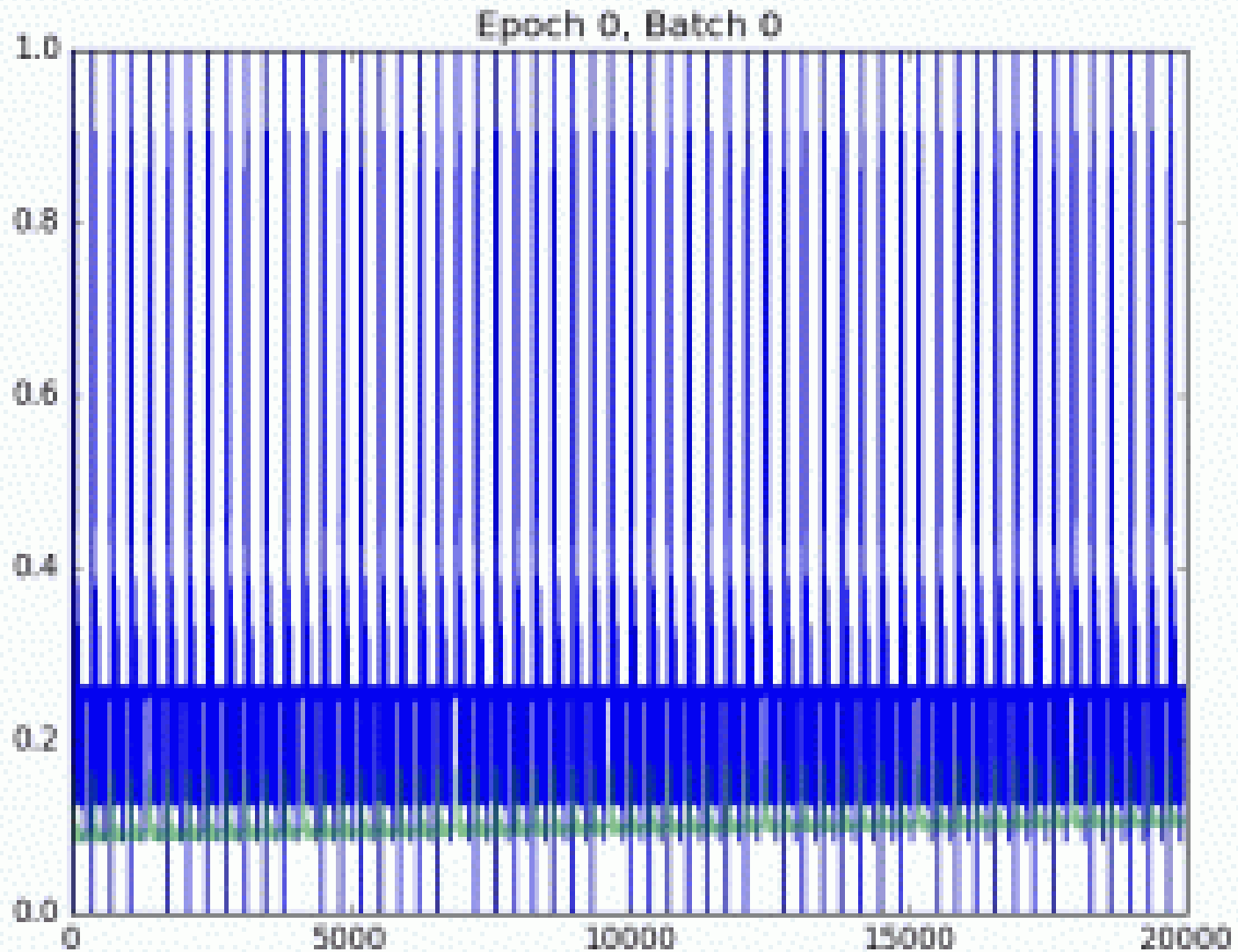
Correlation Coeff. between research parameters



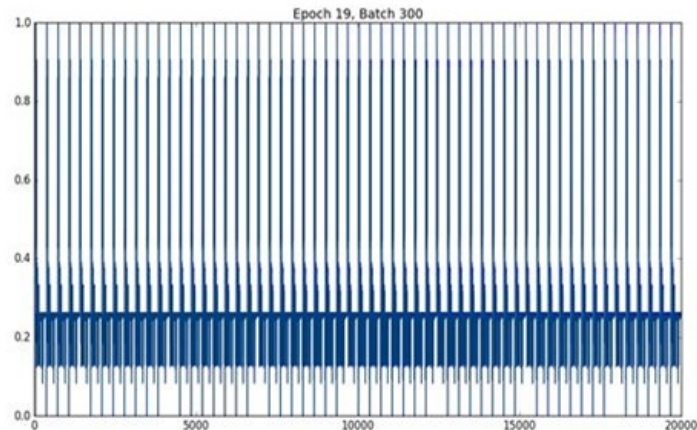
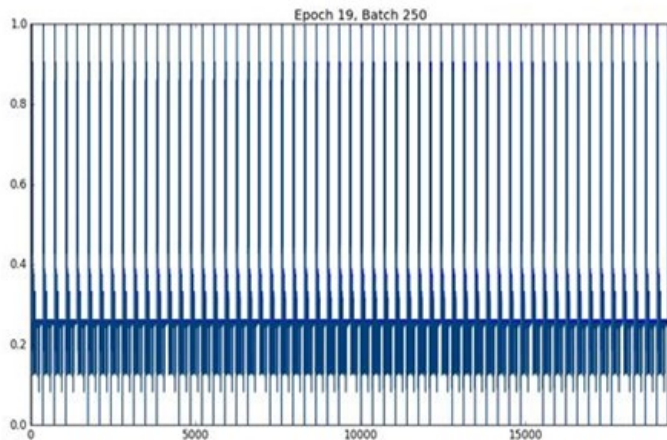
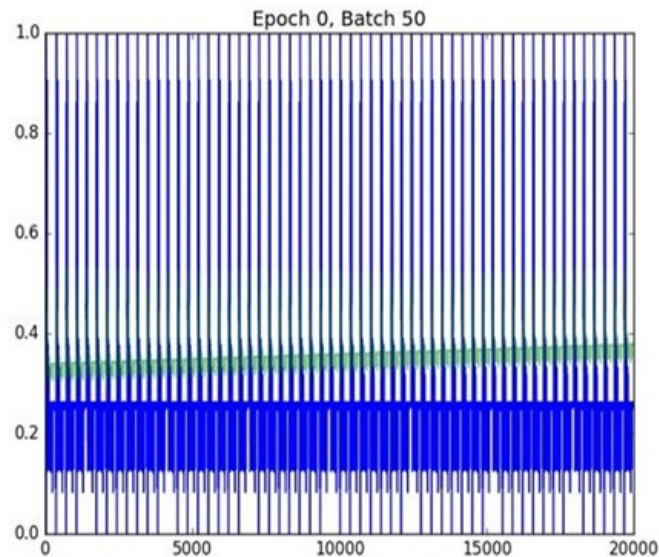
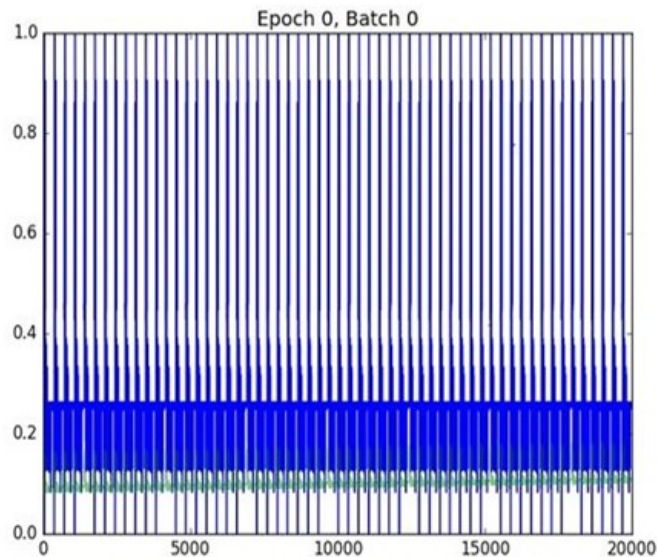
Результаты обучения и тестирования РНС



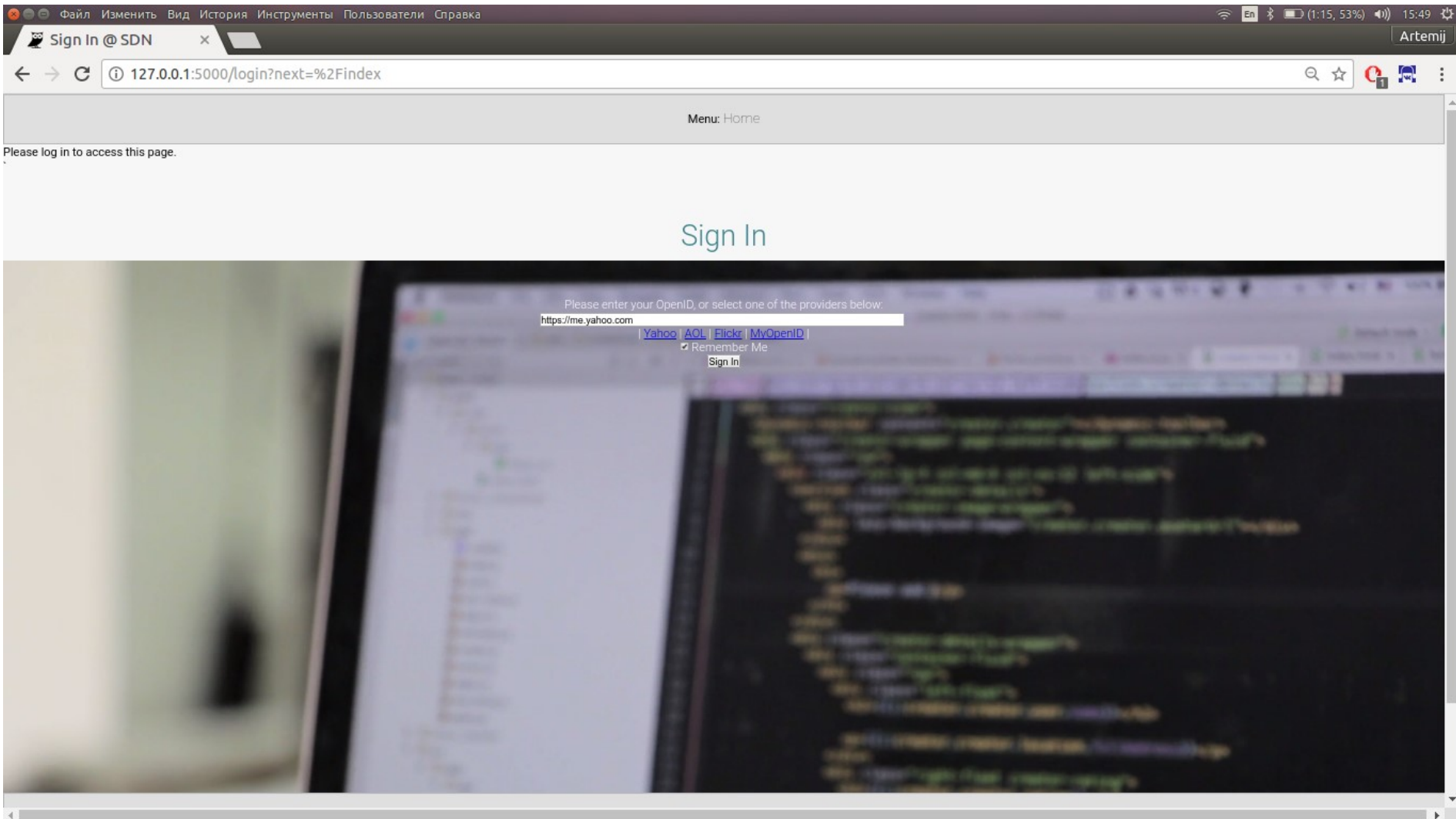
Процесс обучения РНС



Результаты обучения и тестирования РНС



Приложение мониторинга SDN



Ваш логотип

Приложение мониторинга SDN

Home @ SDN | 127.0.0.1:5000/index

Menu: Home | IoT Data Base | Controllers | Switches|Logout |

Engeneer system SDN

Current SDN Topology

Connected controller:

Name:
Opendaylight
Beryllium

ip Address:
192.168.0.3

Vendor:
Opendaylight
Foundation

Version: Beryllium
SR4

Show [10] entries

Switch ID	Switch Vendor	Soft version	Hardware version	MAC Address	IP Address	Buffer size	Flow Table	Number of hosts	Number of IoT's
openflow:1	Cbench - controller I/O benchmark	this is actually software...	version 0.01	00:00:00:00:01:01	192.168.0.6	256	2	hosts	IoT's
openflow:10	Cbench - controller I/O benchmark	this is actually software...	version 0.01	00:00:00:00:0a:01	192.168.0.6	256	2	hosts	IoT's
openflow:1096921035519896	MikroTik	RB2011UIAS	RouterOS 6.35.2	00:8c:34:4f:98:01	192.168.0.2	0	1	hosts	IoT's
openflow:2	Cbench - controller I/O benchmark	this is actually software...	version 0.01	00:00:00:00:02:01	192.168.0.6	256	2	hosts	IoT's
openflow:256040834877923	Nicira, Inc.	Open vSwitch	2.0.2	00:27:02:c9:e3:01	192.168.0.6	256	254	hosts	IoT's
openflow:3	Cbench - controller I/O benchmark	this is actually software...	version 0.01	00:00:00:00:03:01	192.168.0.6	256	2	hosts	IoT's
openflow:4	Cbench - controller I/O benchmark	this is actually software...	version 0.01	00:00:00:00:04:01	192.168.0.6	256	2	hosts	IoT's
openflow:5	Cbench - controller I/O benchmark	this is actually software...	version 0.01	00:00:00:00:05:01	192.168.0.6	256	2	hosts	IoT's
openflow:532771570470800	MikroTik	RB2011UIAS	RouterOS 6.35.2	00:8c:34:4f:90:01	192.168.0.2	0	1	hosts	IoT's
openflow:6	Cbench - controller I/O benchmark	this is actually software...	version 0.01	00:00:00:00:06:01	192.168.0.6	256	2	hosts	IoT's

Showing 1 to 10 of 14 entries

Copyright © 2017 Developer Volkov Artem / SDN Laboratory/ sdnlab.ru / Saint-Petersburg state University of Telecommunications named by M.A. Bonch-Bruевич

Приложение мониторинга SDN

Switches @ SDN | 127.0.0.1:5000/switches | Artemij

Menu: Home | IoT Data Base | Controllers | Switches | Logout |

Engineer system SDN

Current SDN Topology (only switches)

Load Graph:

Links Load on current Topology
Data from controller

Link	Approximate Load (Mbits)
link_1	0.1
link_2	1.1
link_3	1.1

```

graph TD
    S1[openflow:1096821035519896] --- link_1 --- S2[openflow:532771570470800]
    S1 --- link_2 --- S3[openflow:814246547181460]
    S2 --- link_3 --- S3
    
```

*	0	0	230	[CONTROLLER]
*	0	0	223	[1]
*	0	0	230	[*, *, *, *]
*	0	0	223	[3]
*	92	10396	230	[CONTROLLER]
*	0	0	230	[CONTROLLER]
*	2	130	136	[3]
*	509	73300	224	[1, 'CONTROLLER']
*	2	126	136	[1]

Show 10 entries | Search:

openflow:532771570470800	#UF\$TABLE*1-10	0	*	*	*	*	*	0	0	250	[*, *, *, *]
openflow:532771570470800	#UF\$TABLE*1-12	100	*	*	*	0x88cc	*	*	0	0	250 [CONTROLLER]
openflow:532771570470800	#UF\$TABLE*1-11	5	*	*	01:80:C2:00:00:02	0x8809	*	*	0	0	250 [CONTROLLER]
openflow:532771570470800	#UF\$TABLE*1-14	2	openflow:532771570470800:4	*	*	*	*	*	14	840	246 [CONTROLLER, '1', '2', '3']
openflow:532771570470800	#UF\$TABLE*1-13	100	*	*	*	0x88cc	*	*	97	11055	250 [CONTROLLER]
openflow:532771570470800	#UF\$TABLE*1-9	0	*	*	*	*	*	*	0	0	250 [*, *, *, *]
openflow:532771570470800	#UF\$TABLE*1-16	2	openflow:532771570470800:3	*	*	*	*	*	8	1024	246 [CONTROLLER, '4', '2', '1']
openflow:532771570470800	#UF\$TABLE*1-27	2	openflow:532771570470800:1	*	*	*	*	*	0	0	224 [4, 'CONTROLLER', '2']

Showing 1 to 25 of 43 entries

Previous 1 2 Next

Заключение

- На базе лаборатории были разработан метод идентификации IoT трафика в сетях SDN с целью его мониторинга и управления, а также для проведения тестирования .
- Было разработано приложение для SDN-контроллера для апробации предложенного метода, а также реализации мониторинга сети SDN.
- В результате апробации была доказана работоспособность предложенного метода. Вероятность правильной идентификации трафика ИВ составляет 99,97%
- Разработано приложение мониторинга сети SDN под контроллер Opendaylight
- Разработаны модель и метод мониторинга и управления сетевой инфраструктурой на базе машинного обучения.
- Определены критерии отражающих условия масштабирования ресурсов контроллера программно-определяемой сети.
- На базе предложенных модели и метода разработано программное обеспечение для решения задач аналитики, корреляции и прогноза развития SDN сетей. Проведен натурный эксперимент, подтверждающий достоверность предлагаемого метода.
- Обнаружена квазилинейная зависимость сетевых параметров SDN сети и аппаратных характеристик SDN-контроллера. Коэффициенты корреляции параметров оказались в интервале 0,89-0,98.
- Определен коэффициент среднеквадратической ошибки при обучении РНС, который составил 0,00000454 и 0,0000153 для обучающей и тестовой выборки соответственно.
- ПО РНС позволило прогнозировать показатели значений пакетов на уровне управления SDN сетью на 5,5 часов вперед, при скорости обучения сети 0,4 часа.

Thank you!

ammarexpress@gmail.com

