

Разработка интеллектуальной системы взаимодействия обучающихся с
оборудованием модельных сетей с целью повышения качества проводимых
лабораторных работ

Белов А. С., Киструга А. В., Маслюхин С. М.

Как система обучения

- Разнообразные курсы по инфокоммуникационным технологиям
- Работа с “живым” оборудованием, а не его эмуляцией
- Статистика пройденных работ, экспорт результатов и отчетов
- Адаптивное обучение на базе сервиса рекомендаций
- Вариативность заданий и градация сложности
- Библиотека учебных материалов
- Кабинет преподавателя
- Удаленный доступ
- Локализация

Как универсальная платформа

- Универсальные программные интерфейсы и контракты
- Интеграция с другими продуктами и сервисами
- Редакторы компонентов (модельных сетей, курсов и работ)
- Горизонтальное масштабирование “железных” ресурсов
- Нет привязки к языку разработки
- Декларативное описание контента

Как инструмент администрирования

- Отображение состояния фрагмента сети и ее компонентов
- Удаленный терминал и элементы управления

Процесс прохождения типовой лабораторной работы

The screenshot displays the Qnet+ 4.6.2 interface with a network diagram and a terminal window. The interface includes a navigation menu on the left, a main workspace for the network diagram, and a terminal window at the bottom.

Network Diagram: Shows a topology with nodes: ClSoho, FSrv, IPSrv, MiTkGW, and MiTk. The IPSrv node is circled in red.

Terminal Window: Shows the following output:

```
>_ Терминал #1 >_ Терминал #2 Библиотека История
Disconnect ClientSoho
ClientSoho connection...
ClientSoho successful connection
administrator@mikclient01:~$
```

Task List (Left Panel):

- 1. Подключение к маршрутизатору (circled in red)
- 2. Информация о системе маршрутизатора (circled in red)
- 5. Определите имя маршрутизатора (circled in red)

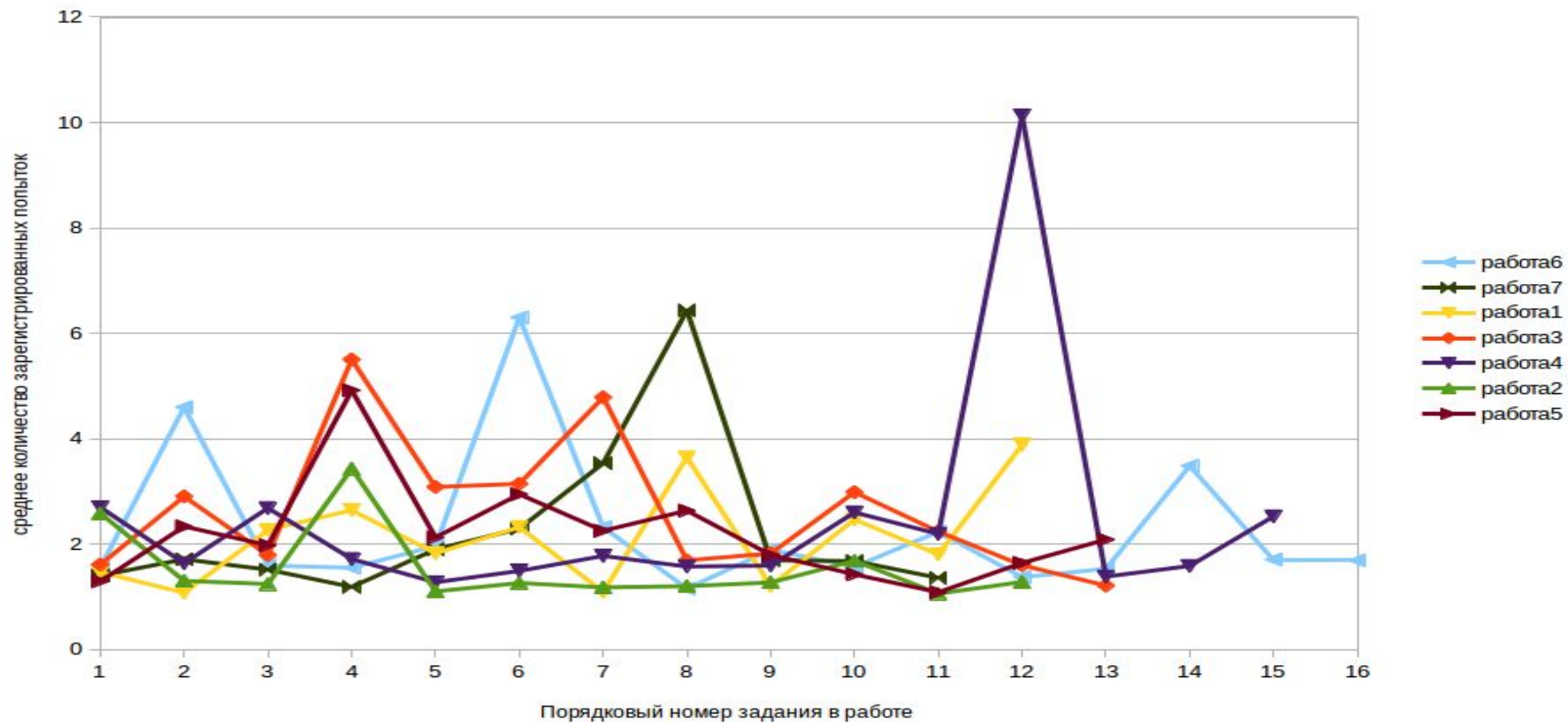
Terminal Input: The input field shows "Определите имя маршрутизатора" (circled in red).

Terminal Output: The output shows "ClientSoho connection..." and "ClientSoho successful connection" (circled in red).

Terminal Buttons: "Disconnect" and "ClientSoho" buttons are visible (circled in red).

Terminal Status: "[session: 0:27] ClientSoho ready" is shown at the bottom.

Среднее количество попыток при выполнении работ



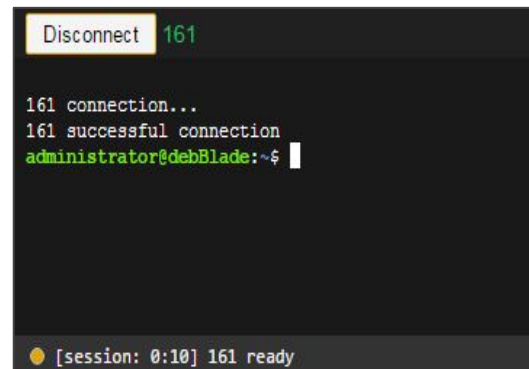
Проблемы взаимодействия с оборудованием

Подготовка модельной сети

- Оборудование различных вендоров и ОС должно быть представлено на схеме сети единообразно
- Удаленный доступ к оборудованию может осуществляться с помощью разных протоколов
- Оборудование может быть виртуальным

Процесс выполнения практической работы обучающимся

- Необходимость в лишней работе с консолью (или браузером)
- Размытие предмета обучения в рамках текущей работы
- Побочные эффекты при оценке результата проделанной работы
- Доступ к технической реализации некоторых механизмов
- Повышенный риск привести модельную сеть в нерабочее состояние
- Безопасность



```
Disconnect 161
161 connection...
161 successful connection
administrator@debBlade:~$
```

[session: 0:10] 161 ready

Использование псевдонимов команд и шаблонов

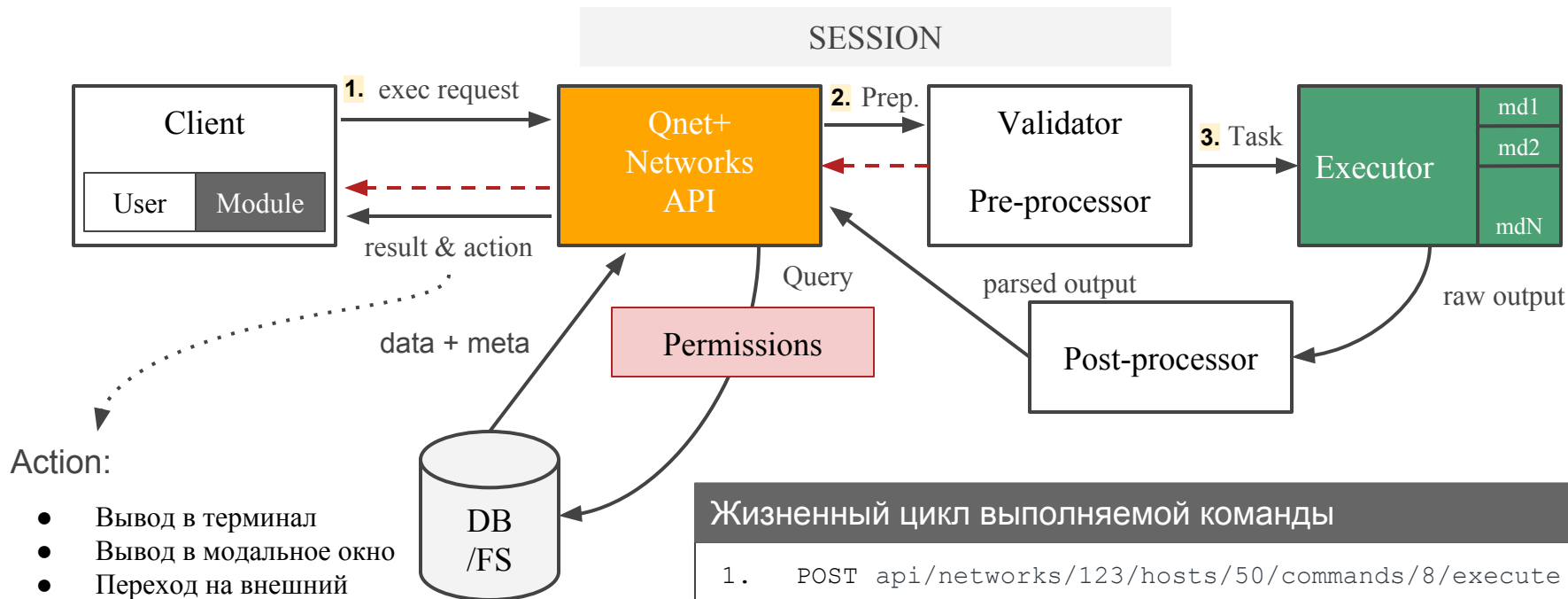
Псевдоним команды - короткое имя, характеризующее действие команды. Один псевдоним может быть использован для неограниченного числа команд, так как он отражает ЧТО нужно сделать, а не КАК. Если логика работы команды не зависит от **параметров**, то команда определяется **шаблоном**.

- Возможность использования пользовательских параметров, атрибутов целевого хоста и устройств модельной сети, данных варианта текущей работы, прочей мета-информации о задании или работе
- Сохранение промежуточных результатов для использования в других командах в рамках сессии
- Ограничение доступа к команде по профилю и по времени
- Группировка команд по различным критериям использования, контекст доступности
- Вариативная обработка результата выполненной команды

Типы выполняемых команд

- Выполняемые на оборудовании модельной сети (ifconfig, uname, reboot и тд)
- Выполняемые на самом сервере, как правило по отношению к целевому хосту (ping)
- Переход по внешней ссылке (стороннее приложение или ресурс)

Общая схема выполнения команды интеллектуального доступа



Action:

- Вывод в терминал
- Вывод в модальное окно
- Переход на внешний или внутренний ресурс
- Отложенная обработка результата
- ...

Жизненный цикл выполняемой команды

1. `POST api/networks/123/hosts/50/commands/8/execute`
`params: { count: 5 }`
2. `EXEC ping {{th:host}} -c {{p:count}}`
`meta: {Object}, context: {Object}`
3. `self_execute ping 10.0.0.1 -c 5`

Привязка выполняемых команд к оборудованию

Пользовательские параметры команды

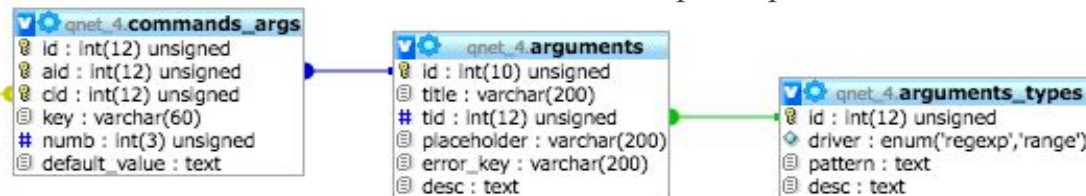
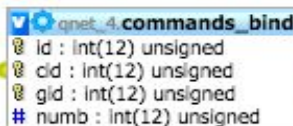
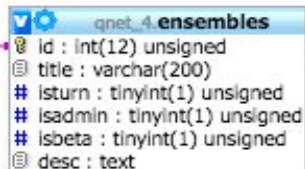


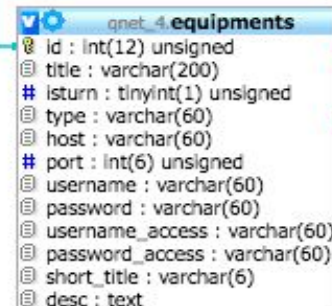
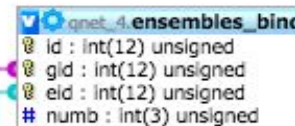
Таблица команд



Группы команд

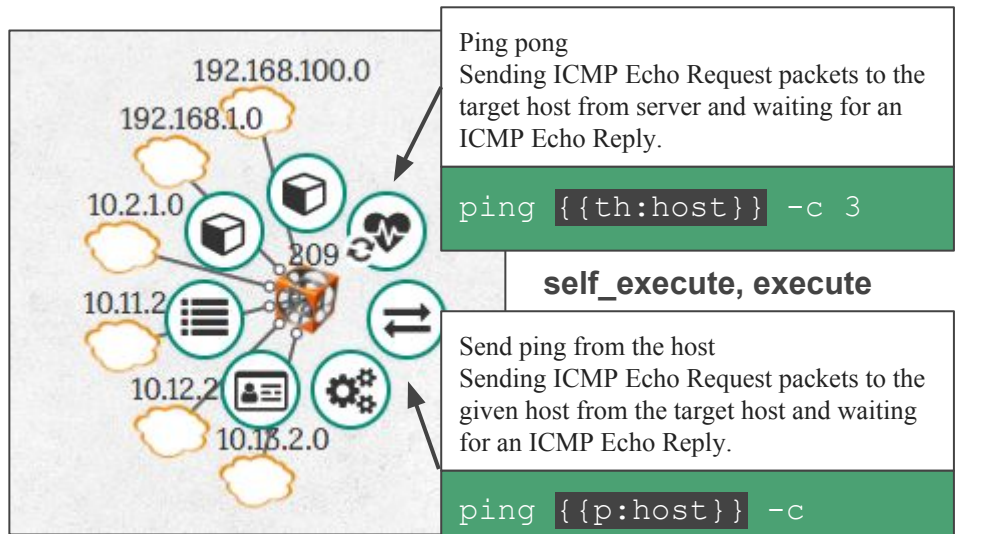


Привязка к оборудованию



Уточнение контекста команды

Пример 1: Отправка ICMP Echo Request



Ping target host

```
PING 172.16.117.209 (172.16.117.209) 56(84) bytes of data.  
64 bytes from 172.16.117.209: icmp_req=1 ttl=64 time=0.067 ms  
64 bytes from 172.16.117.209: icmp_req=2 ttl=64 time=0.045 ms  
64 bytes from 172.16.117.209: icmp_req=3 ttl=64 time=0.050 ms
```

```
--- 172.16.117.209 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 1998ms  
rtt min/avg/max/mdev = 0.045/0.054/0.067/0.009 ms
```

209
172.16.117.209

Access to the host denied by the network settings.

General

- Ping pong
- **Send ping from the host**
- Traceroute

Linux Debian

- Get interface configuration
- Get system name
- Get list directory contents
- Echo known system name

Squitch server ⓘ

- Open squitch on this server
- Start SIP INVITE scenario

Scenario OSPF ⓘ

- Start scenario

Send ping from the host

Target host:
localhost

Packets amount:
5

Execute

```
ping 172.16.117.209 -c 3
```

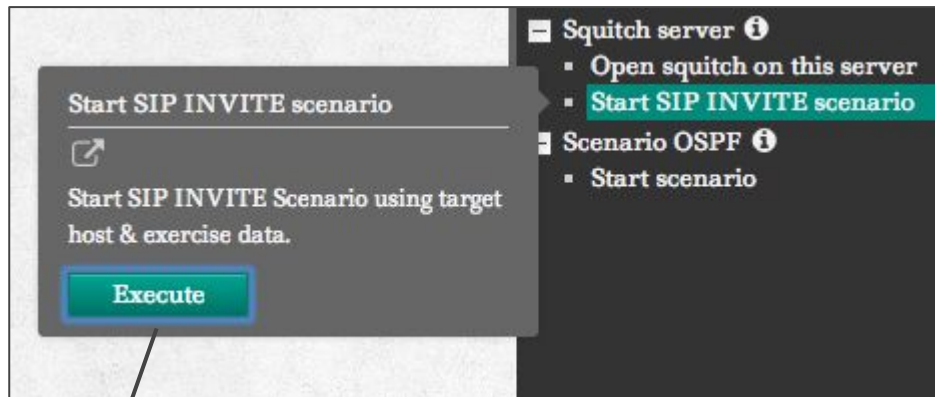
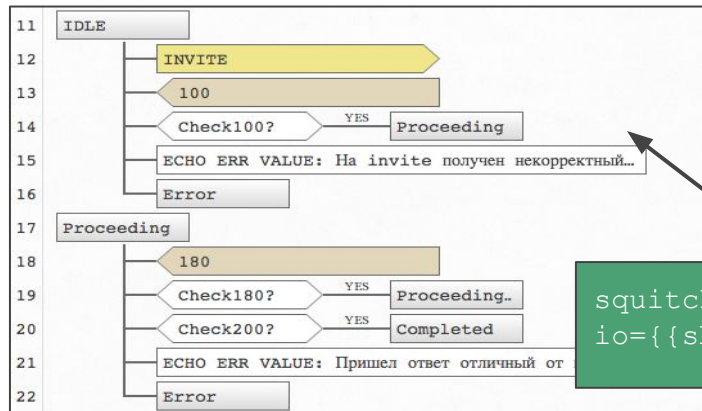
```
PING 172.16.117.209 (172.16.117.209) 56(84) bytes of data.  
64 bytes from 172.16.117.209: icmp_req=1 ttl=64 time=0.067 ms  
64 bytes from 172.16.117.209: icmp_req=2 ttl=64 time=0.045 ms  
64 bytes from 172.16.117.209: icmp_req=3 ttl=64 time=0.050 ms
```

```
--- 172.16.117.209 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 1998ms  
rtt min/avg/max/mdev = 0.045/0.054/0.067/0.009 ms
```

Пример 2: Выполнение SIP-сценария через платформу Squitch

1. Через браузер или иконку на рабочем столе **открыть** Squitch
2. Выбрать нужную **дисциплину**, в дереве испытаний найти нужный **сценарий**
3. Заполнить требуемые **параметры** сценария и только потом запустить

- Пользователь не имеет доступа ко всей функциональности внешнего приложения
- Не использует внешние инструменты
- Не ответственен за корректность параметров



link

```
squitch-dev/?i={{th:opts:id}} &scenar  
io={{sbj:name}}
```



API

Пример 3: Использование tshark и file-сервера

1. На целевом хосте: `>filename.pcap; chmod a+w filename.pcap ; sudo tshark -i eth1 -l -a duration:360 -w filename.pcap>/dev/null 2>&1 &`
2. По завершению: `scp file1.pcap administrator@172.16.116.5:labs/ + password`
3. На file-сервере: **скачать файл** и открыть через wireshark

```
/webshark-dev/main/index.php?selecte  
d_pcap_filepath={{sbj:filename}}{{IN  
DEX+1}}.pcap&filepath=dumps/ {{sbj:fo  
lder}}&hide_other_files=true
```

- Пользователь не задает имя файла, путь к файлу, не знает паролей
- Не может случайно или умышленно запустить tshark дважды
- Не тратит время на ввод сложных и ненужных в контексте работы команд
- А нужен ли File-сервер на схеме вообще?



```
>{{sbj:filename}}{{INDEX+1}}.pcap ;  
chmod a+w  
{{sbj:filename}}{{INDEX+1}}.pcap ;  
sudo tshark -i eth1 -l -a  
duration:{{p:duration}} -w  
{{sbj:filename}}{{INDEX+1}}.pcap
```

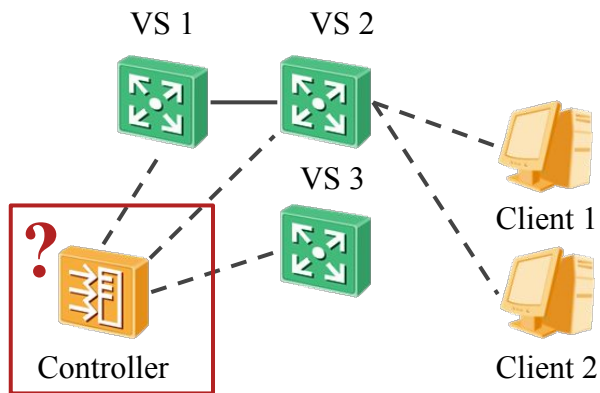
```
expect -c 'spawn scp -o  
UserKnownHostsFile=/dev/null -o  
StrictHostKeyChecking=no  
{{sbj:filename}}{{INDEX+1}}.pcap  
administrator@{{FSADDR}}:Dumps/{{sbj  
:folder}}/; expect "assword:"; send  
"password\n\r"; interact'
```

Пример 4: Работа с виртуальными коммутаторами через контроллер (SDN/OVS)

- Виртуальные коммутаторы **не имеют IP адреса** и создаются **динамически**, таким образом не могут быть представлены как устройства в терминах базы данных
- Невозможно добавить подобные устройства на **схему модельной сети** стандартными средствами
- Управление этими устройствами осуществляется только через **контроллер** (гипервизор)

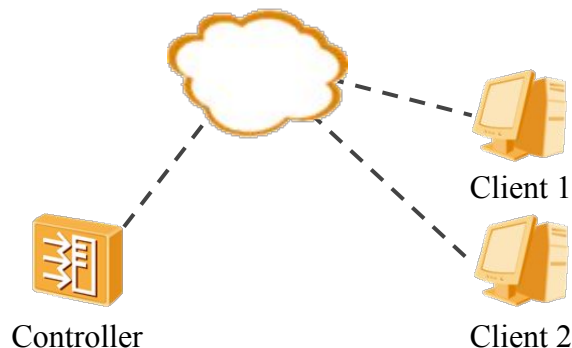
I

Отрисовка виртуальных устройств (расширение модуля построения схемы), управление проксируется через контроллер фоном.



II

Виртуальные устройства не отображаются на схеме, управление через контроллер с возможностью выбора свитча.



Дальнейшие направления разработки

- Real-time для команд интеллектуального доступа
- Индикация на схеме проходящего трафика в модельной сети
- Интеграция с другими сервисами (Nester, трафик-генераторы)
- Редактор интеллектуальных команд для модераторов курсов
- Улучшение алгоритмов оценки прохождения заданий
- Интеллектуальный веб-терминал
- Внешний API для сторонних продуктов
- Сервис мониторинга используемых в работах сетевых ресурсов
- Использование схемы стенда и интеллектуального доступа как инструмент мониторинга и управления сетями