Packet Tracer - Explore the Smart City (перевод с англ.)

Topology



1. Objectives
* **Explore Packet Tracer as a Prototyping Tool**
1. Background / Scenario

В этом упражнении у вас будет возможность изучить прототип системы IoT на основе Packet Tracer, предназначенный для того, чтобы пользователь мог удаленно проверить, открыта ли дверь гаража. Вам предлагается свободно исследовать прототип, чтобы лучше понять технологии, используемые в системе.

1. Required Resources
* На компьютере должна быть установлена программ Packet Tracer 7.1 или более поздней версии.
* Легально скачать Cisco Packet Tracer можно по адресу:
**https://www.netacad.com/courses/packet-tracer**

**Прим.:** Необходимо пройти авторегистрацию в рамках курса для самостоятельного обучения работе в среде Cisco Packet Tracer.

1. The Topology
	* 1. The Network

Топология сети предназначена для моделирования дома, который подключен к Интернету через поставщика услуг Интернета (ISP) на основе DSL. Беспроводной маршрутизатор обеспечивает беспроводной доступ ко всем домашним устройствам (настольные компьютеры, ноутбуки, смартфоны и т. Д.).

Беспроводной маршрутизатор также подключен к глобальной сети через модем DSL, который, в свою очередь, подключается к Интернету. Обратите внимание, что модем DSL предоставляется Интернет-провайдером.

Устройство SBC представляет собой компьютер Raspberry Pi и подключено к двери гаража (**Garage Door**). Поскольку дверь гаража не имеет собственного интеллекта или подключения к Интернету, SBC используется для восполнения этого пробела. Как и другие домашние устройства, SBC подключается к домашней беспроводной сети через домашнюю беспроводную сеть.

**Home PC** и **Home Laptop** были добавлены в качестве домашних устройств, которые подключены к Интернет. Любое из этих устройств также может быть использовано для доступа к системе.

**Public WiFi Router** добавлен для имитации публичного интернет-соединения в города. Это моделирование работы wireless hot spots, установленных в библиотеках, кафе, метро, на остановках общественного транспорта и т.д. Ключевым моментом здесь является то, что пользователь также будет иметь подключение к Интернету.

**Smartphone** представляет пользователя, когда он или она вдали от дома. В этом случае доступ к Интернету будет осуществляться через этот смартфон. Обратите внимание, что пользователь также может подключаться к системе с любого другого устройства, если оно подключено к Интернету.

**Прим.**: Python, используемый в PT, является интерпретатором Python to JavaScript с открытым исходным кодом, который не обновляется до Python 3.0. По этой причине могут быть небольшие различия в синтаксисе кода, наблюдаемого в PT, и кода в устройствах, использующих Python 3.

* + - 1. Click the **SBC**.
			2. Click the Programming Tab. На открывшейся левой панели отображается имя файла, **main.py**.
			Это программа, которая будет работать в SBC от имени **Garage Door**.
			3. Packet Tracer отображает фактический код, сохраненный в **main.py**, в правой части окна. Это важно, потому что у Вас есть возможность редактировать, останавливать и запускать программу.
			4. Click the **Run** button для активации программы. Опишите реакцию системы.

|  |
| --- |
| текст |

* + - 1. Этот код, написанный на Python, отвечает за периодическую передачу статуса гаражных ворот на сервер в Интернете. Если дверь закрыта, код отправляет 0. Если дверь открыта, 1 отправляется на интернет-сервер.

Проанализировав код, ответьте на следующие вопросы:

Как часто обновления статуса двери отправляются на интернет-сервер?

|  |
| --- |
| текст |

Определите IP-адрес и TCP порт назначенного сервера?

|  |
| --- |
| текст |

Какие имена переменных используются для хранения IP-адреса и номера порта TCP сервера?

|  |
| --- |
| текст |

Какие модули Python были импортированы в код?

|  |
| --- |
| текст |

1. Using the System

Чтобы использовать систему, рассмотрим следующий сценарий:

*Пользователь покидает дом и останавливается в кафе по пути на работу. Во время остановки пользователь пытается вспомнить, была ли дверь гаража закрыта или нет. Поскольку проверяемая IoT система уже установлена, пользователь может быстро получить доступ к адресу интернет-сервера через веб-браузер своего телефона, чтобы проверить состояние двери.*

* + 1. Opening the Status Page From the Smartphone
			1. Click the Smartphone и выберите закладку **Desktop**.
			2. В закладке **Desktop** запустите режим **Web Browser**.
			3. Введите в строку поиска URL **www.connthings.example**. Это IP-адрес интернет-сервера, используемого для сбора данных, отправленных SBC.
			4. Страница, которая открывается в смартфоне, должна отображать зеленую полосу, сообщающую, что дверь гаража закрыта (garage door is shut).
			5. Когда на смартфоне все еще открыта страница состояния, откройте дверь гаража, удерживая нажатой клавишу ALT и щелкая дверь. Дверь должна открыться.

Как это отображается на смартфоне?

|  |
| --- |
| текст |

SBC, подключенный к двери, отправляет обновления на сервер, который, в свою очередь, обновляет локальный файл со статусом двери. Когда пользователь подключается к серверу через веб-браузер, сервер представляет веб-страницу, созданную на основе состояния двери, сохраненной на диске.

Система предназначена для того, чтобы любое устройство, подключенное к Интернету, могло ее использовать. Попробуйте открыть страницу состояния двери с домашнего компьютера Home PC и с домашнего ноутбука Home Laptop. Проверьте как это работает.

1. Challenge Questions
	* + 1. Разверните Интернет-облако и изучите его содержимое. Попробуйте найти DNS-сервер и веб-сервер, которые получают обновления от SBC.
			2. Найдите и исследуйте устройство **Multilayer Switch**. Какова его функция в этом прототипе?

|  |
| --- |
| текст |

* + - 1. Добавьте больше устройств в домашнюю сеть и попробуйте получить от новых устройств доступ к системе.
			2. Проверьте доступность ресурса **http://209.165.201.3**.
1. Reflection

Прототип системы, представленной здесь, является просто доказательством концепции, но может использоваться для представления и даже уточнения некоторых аспектов проекта. По своей природе прототипы Packet Tracer очень недороги, поскольку позволяют тестировать проекты, требующие сложных топологий сети. Кроме того, поддержка Python в Packet Tracer облегчает улучшение и в конечном итоге переносит любой код на реальные устройства после завершения этапа тестирования.

**Прим.:** данная лабораторная работа является частью курса IoT Fundamentals: Connecting Things