Интеллектуальный кампус представлял собой более комплексное моделирование IoT по сравнению с двумя предыдущими упражнениями для умного дома, как по сети, так и по компоновке IoT, которые на самом деле были более сложными, чтобы продемонстрировать более глубокое взаимодействие между устройствами IoT, но также предоставить студентам больше возможностей в будущем. расширение упражнений. Smart-Campus моделировал университетский камуфляж, где, наряду с традиционными сетями школ и жилых помещений, сеть IoT позволяла подключать различные устройства IoT, распределенные по территории кампуса. Примеры управления доступом RFID и интеллектуального решения для полива спортивных полей были включены в симуляцию.

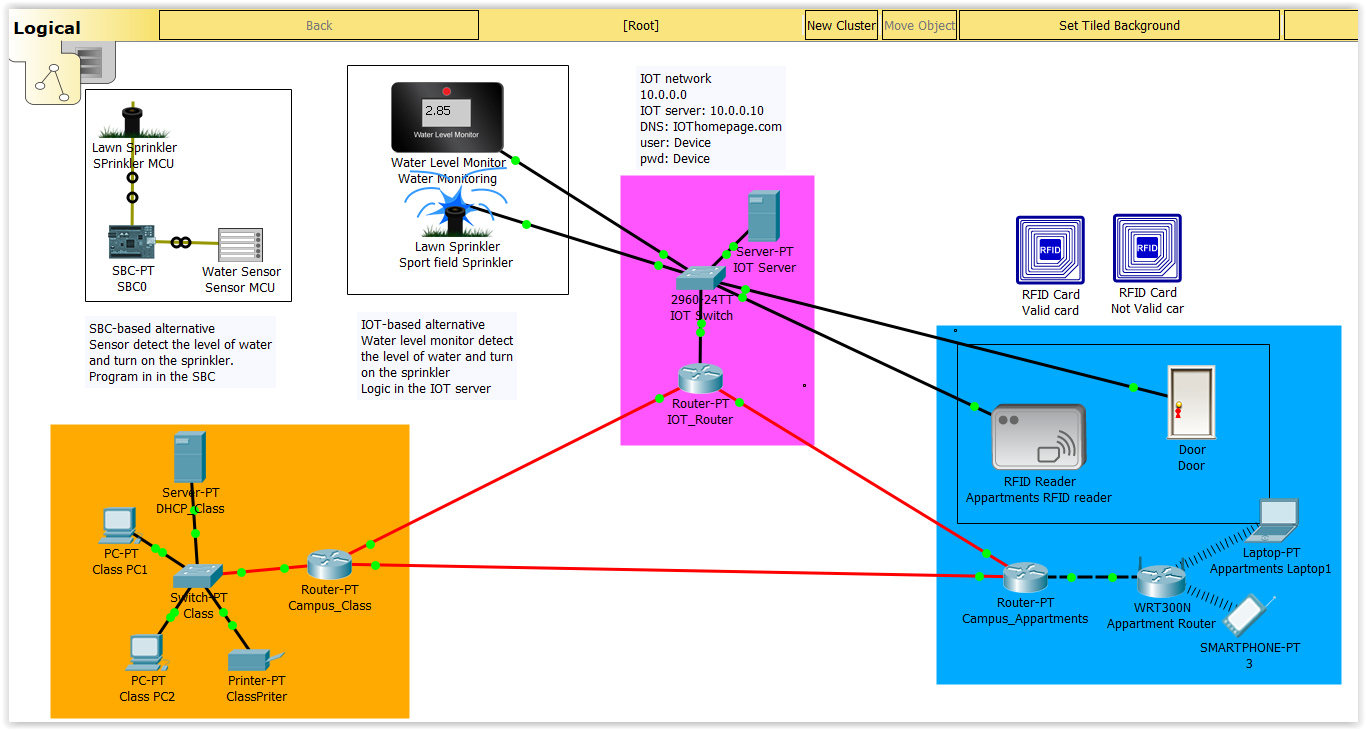


Рисунок 37 - Схема Cisco Packet Tracer для моделирования Smart-Campus

Расположение сети

Как кратко упомянуто, схема сети в этом упражнении была более сложной по сравнению с предыдущими моделями и включала в себя: магистральную маршрутизаторную сеть, традиционную проводную сеть на базе коммутатора, беспроводную локальную сеть для жилых зданий и выделенную сеть IoT, также основанную на на переключателе.

Магистральная сеть была создана с использованием трех взаимосвязанных маршрутизаторов. Каждый маршрутизатор имел соединение с двумя другими для создания резервной инфраструктуры, которая могла бы противостоять сбоям магистралей между маршрутизаторами.

Чтобы представить ареалистическую сеть, в которой маршрутизаторы физически размещались в разных зданиях кампуса, вместо традиционных прямых медных кабелей использовалась оптическая проводка Fast-Ethernet.

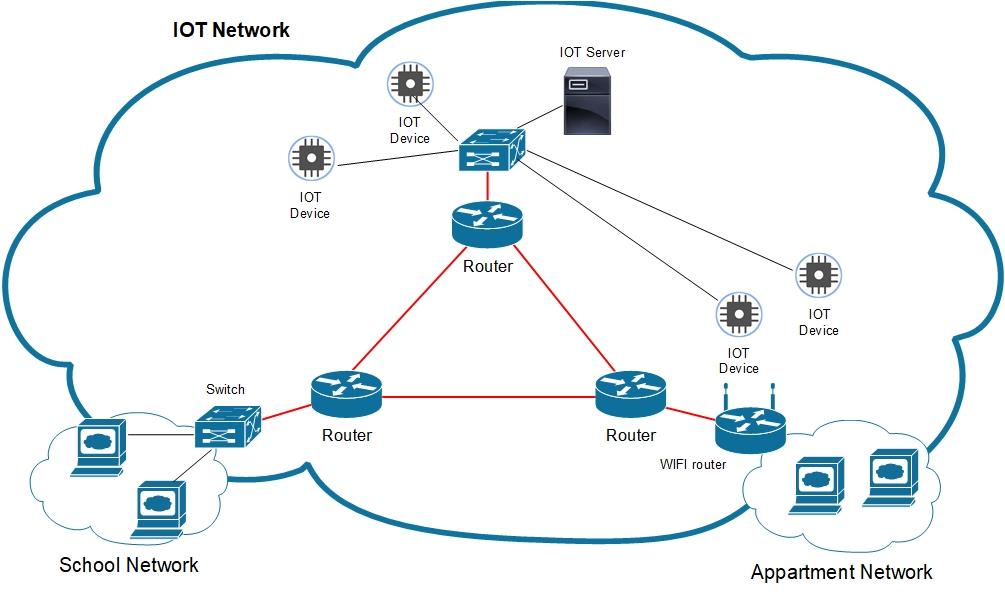


Рисунок 38 - Топология сети Smart-Campus

Для упрощения маршрутизации между магистральными устройствами, чтобы обеспечить полное соединение между тремя сетями использовался базовый протокол маршрутизации (RIP) в конфигурации роутера. RIP - это очень простой и старый протокол маршрутизации, который периодически разделяет таблицу маршрутизации между устройствами. В реальных сложных сценариях протокол обычно не используется из-за его ограничения масштабируемости, поскольку на самом деле протокол допускает не более пятнадцати сетевых переходов

Тем не менее, из-за своей простоты настройки, RIP был идеальным кандидатом для протокола маршрутизации в

Упражнение Cisco Packet Tracer.

В каждом маршрутизаторе настройка была выполнена с добавлением IP-адресов напрямую подключенных сетей в конфигурации RIP, как показано на рисунке 39, тогда обмен таблицей маршрутизации позаботится о распределении логики маршрутизации между устройством, как показано на рисунках 40 и 41.

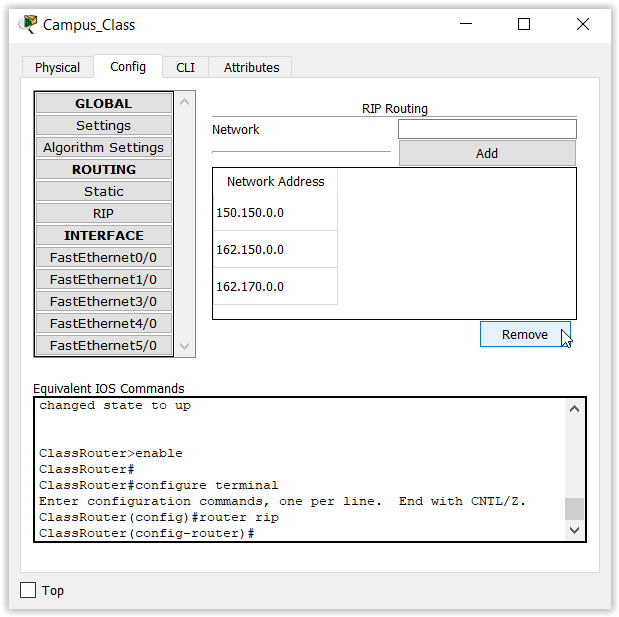


Рисунок 39 - Пример простой настройки RIP в классе маршрутизатора

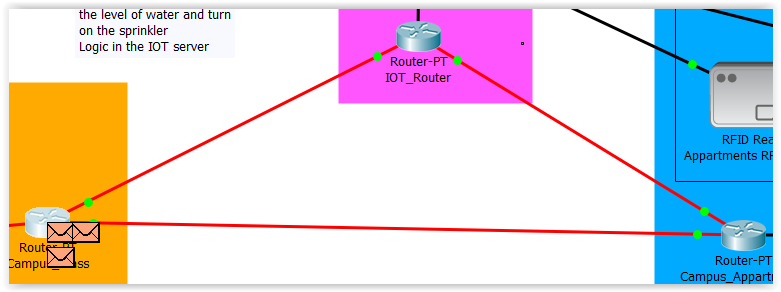


Рисунок 40 - Пример трансляции RIP-сообщений маршрутизатором Classroom

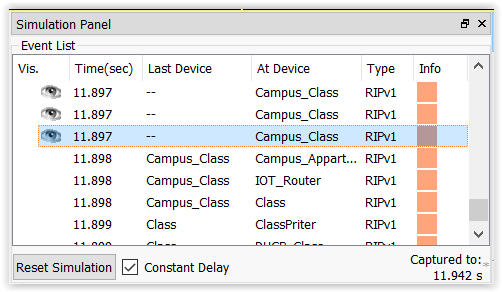


Рисунок 41 - пакет, захваченный во время широковещательной рассылки RIP-сообщений

Наряду с магистральным соединением каждый маршрутизатор также был подключен к одному из трех подсети: сеть класса здания, сеть многоквартирного дома и сеть IoT.

Все три сети были физически разделены путем помещения их в собственный выделенный физический контейнер.

Первая сеть представляла собой простую сеть для эмуляции класса ПК, где два компьютера, один сетевой принтер и сервер были подключены кабелями Ethernet к коммутатору класса. Затем коммутатор был подключен к одному из портов Ethernet маршрутизатора. Функции DHCP выполнялись локальными серверами.

Сеть жилых домов была также простой сетью WLAN, которая имитировала беспроводную связь в жилых домах студентов. В этом случае маршрутизатор WLAN использовался для создания локальной беспроводной сети, затем маршрутизатор был подключен к одному из магистральных маршрутизаторов. Функции DHCP в этой сети также выполнялись маршрутизатором WLAN. Один ноутбук и смартфон были подключены к беспроводной сети.

Последней, но самой важной сетью была сеть IoT. Это была сеть на основе коммутатора, подключенная к третьему магистральному маршрутизатору. Все устройства IoT и сервер IoT были подключены к одному коммутатору. В первоначальной спецификации моделирования IoT предполагалось использовать маршрутизатор WLAN для подключения всех устройств IoT, что делает моделирование ближе к реальности. Однако из-за диапазона покрытия сигнала WLAN и отсутствия беспроводных повторителей в Cisco Packet Tracer, самый дальний IoT-устройства имели очень плохое покрытие, и иногда подключение к IoT-серверу прерывалось и прерывалось. Использование переключателя и медные кабели, возможно, не были полностью реалистичными и не применимы в реальных приложениях,но лучше подходит для этого упражнения Cisco Packet Tracer.

IoT-сервер, в дополнение к IoT-бэкэнду, также использовался в качестве DNS-сервера и DHCP-сервера, выделяющего IP-адрес подключенным IoT-устройствам.

План IoT

Как и в предыдущих автоматизациях IoT, все интеллектуальные устройства были удаленно подключены к серверу IoT, используя одни и те же имя пользователя и пароль. Соединение также было установлено с использованием статического IP-адреса сервера IoT, размещенного в той же сети IoT.

Что касается дизайна и структуры сети Duet и использования RIPprotocol, каждый ПК, ноутбук и смартфон, подключенные к любой сети кампуса, могут получать удаленный доступ к URL-адресу домашней страницы IoT. Сервер IoT, являясь также DNS-сервером, преобразовал URL-адрес домашней страницы в свой статический IP-адрес.

При подключении к домашней странице IoT через браузер и после успешной прохождения аутентификации пользователи IoT могли видеть список подключенных устройств и логику взаимодействия между ними.

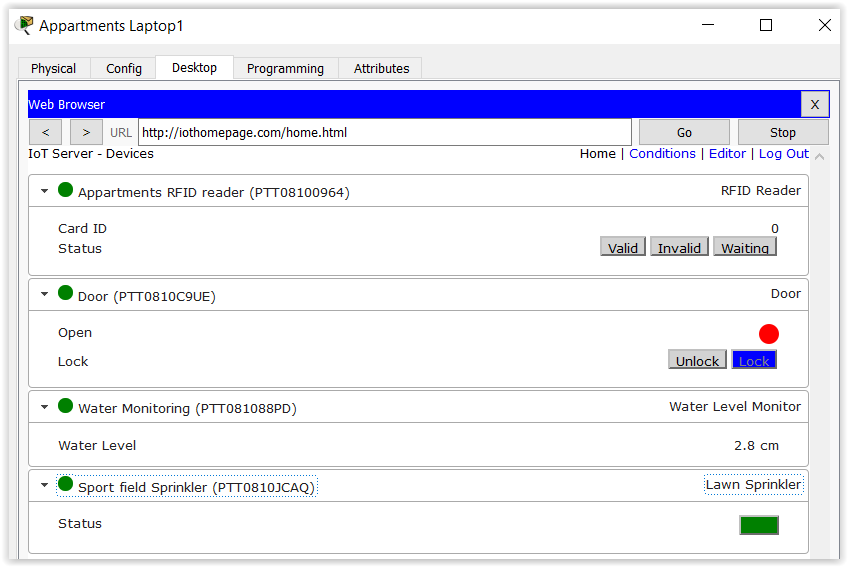


Рисунок 42 - IoT подключенные устройства в моделировании Smart-Campus

Как показано на рисунке 42 выше, в этом упражнении Cisco Packet Tracer были подключены четыре устройства IoT: считыватель RFID, дверь квартиры, водный мониторинг полевого разбрызгивателя sensorandsport.

На рисунке 43 также видна логика, назначенная этим четырем устройствам.

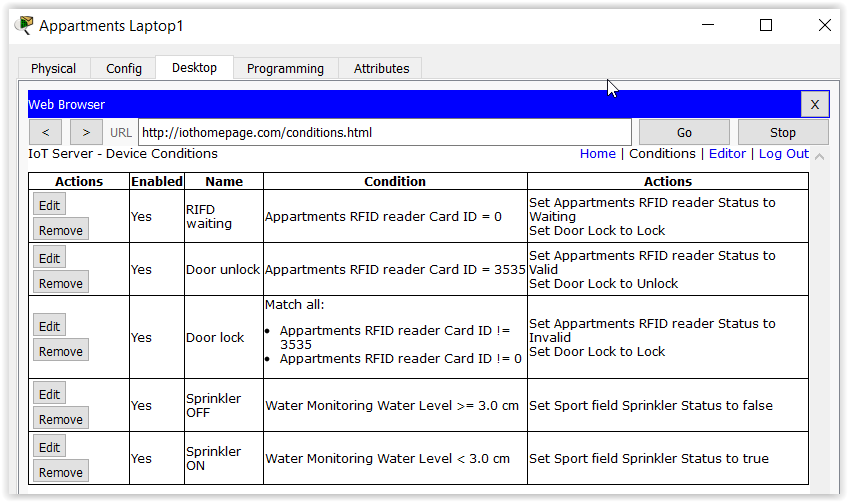


Рисунок 43 - Предварительно установленные условия IoT

Первая автоматизация была направлена на создание решения для контроля доступа IoT RFID для управления доступом в студенческих квартирах с использованием считывателя RFID, пары карт RFID и умной двери.

Концепция была очень проста, когда авторизованная RFID-карта была нанесена на RFID-считыватель, дверь открылась, но если использовалась несанкционированная RFID-карта, дверь оставалась запертой. Для достижения такого сценария параметры карты RFID были изменены путем редактирования вкладки атрибута карты, как показано на рисунке 44 ниже.

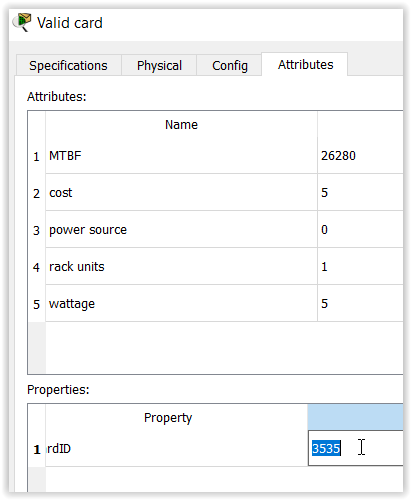
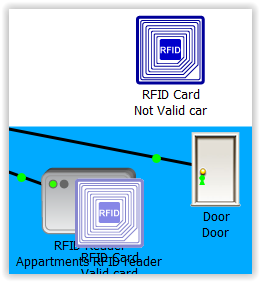
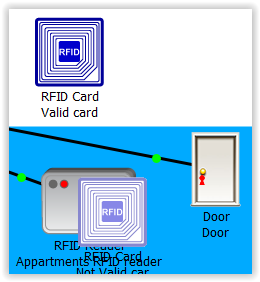


Рисунок 44 - модификация атрибута RFID-карты

В этом случае использовались две карты, одна со значением 3535, а другая со значением 1212. Как показано на рисунке 43, логика авторизации была выполнена на внутреннем сервере IoT, поскольку было установлено условие для разблокировки двери при замене 3535.

Обмен карты был сделан простым перетаскиванием карты на считыватель RFID с помощью мыши. Зеленый значок, появившийся в считывателе, означал, что карта была авторизована, в то же время значок двери стал красным с зеленого. Если использовалась неправильная карта со значением 1212 в примере, значок считывателя оставался красным вместе со значком запертой двери, как видно на рисунке 46.

 ]

Моделирование RFID было создано только с основной целью показать разные сценарии IoT студентам. Однако было ясно, что в более сложных приложениях использование внутренней логики IoT было не самым лучшим вариантом, поскольку при простых условиях было невозможно достичь различных комбинаций карт и уровня доступа.

Следует также упомянуть, что в Cisco Packet Tracer считыватель RFID не всегда работал правильно. При первом запуске симуляции читатель не принимал карты. Студентам рекомендовалось всегда останавливать и перезапускать внутреннюю программу считывателя при первом запуске упражнения. Это было сделано нажатием на устройстве, затем на кнопке продвижения и, наконец, на вкладке «Программирование», когда там можно было остановиться и снова запустите программу, как показано на рисунке 47 ниже.

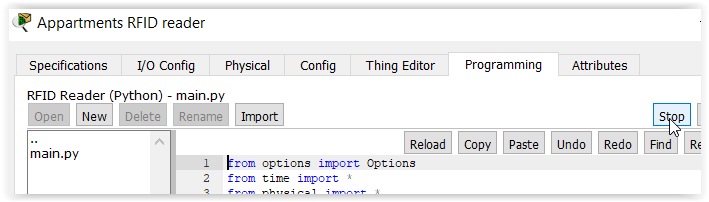


Рисунок 47 - Остановка программы устройства

Во втором случае IoT в этом упражнении была смоделирована интеллектуальная система полива спортивных полей, где датчик обнаружил уровень воды и, в случае низкого уровня рычага, запустил разбрызгиватели воды. Логика видна на рисунке 43.

Чтобы сделать автоматизацию более реалистичной, переменные среды контейнеров для спортивных площадок были изменены, чтобы имитировать дождь в течение нескольких часов в течение дня.

Пример IoT микроконтроллера

Как и в предыдущих двух симуляциях умного дома, в упражнении Cisco Packet Tracer для умного кампуса был добавлен пример микроконтроллера. Имитация воспроизводила тот же случай спринклера IoT при использовании не интеллектуальных устройств.

Входные контакты модуля SBC были подключены через специальный кабель IoT к неумному датчику воды. Выход SBC был затем подключен к разбрызгивателю газона, как показано на рисунке 50 ниже.

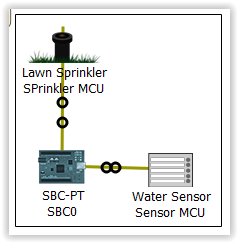


Рисунок 50 - Пример настройки микроконтроллера

Как и в IoT-версии моделирования, датчик воды обнаружил уровень воды и включил дождеватель, если было обнаружено значение ниже порогового значения.

Разница по сравнению со случаем IoT заключалась в том, что логика была определена не на внутреннем сервере IoT, а в пользовательской программе Blockly, хранящейся в самом SBC.

Другое небольшое отличие заключалось в том, что используемый датчик воды возвращал число от 0 255, это значение было затем отображено в программе SBC в диапазоне от нуля до двадцати сантиметров.

Также в этом случае переменные среды дождя были изменены, чтобы выпадать дождь в течение дня.

Логика программы Blockly, запущенной на SBC, видна в Приложении 1.

