



دانشکده مهندسی  
کامپیوتر

بسمه تعالی

دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

درس مبانی اینترنت اشياء

نیمسال دوم سال تحصیلی ۹۸-۹۹

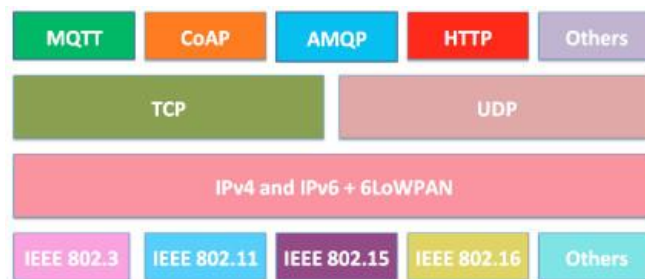
تمرین پیاده سازی سری چهارم



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)

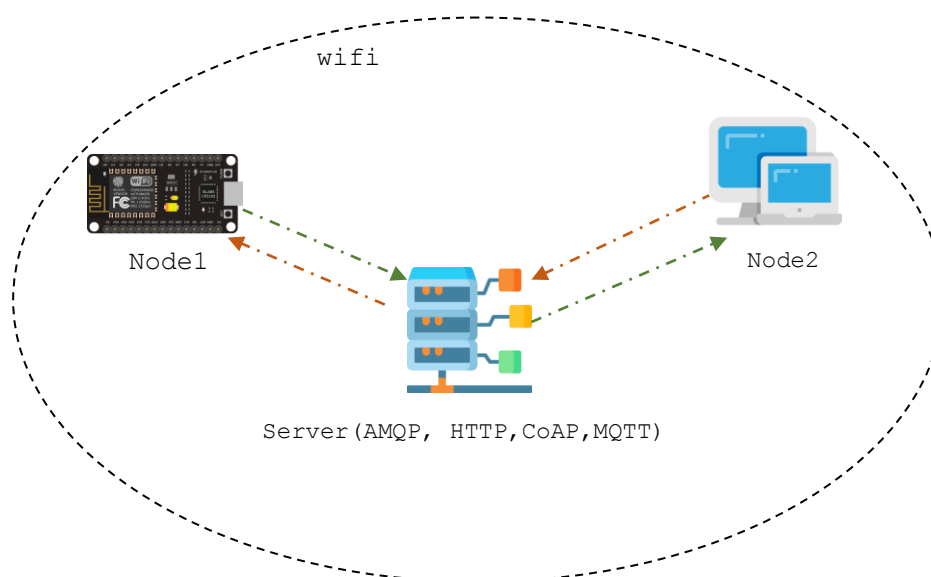
## شرح تمرین

هدف از این تمرین، آشنایی بیشتر و بررسی پروتکل های AMQP, HTTP, CoAP, MQTT است. به طور کلی اینترنت اشیا دارای تعدادی پروتکل در لایه های مختلف می باشد (شکل ۱)، که هر کدام دارای کاربرد خاصی بوده و ویژگی های منحصری به فردی دارند.



شکل ۱- پشته پروتکل در اینترنت اشياء

بنابراین در این تمرین به پیاده سازی و تبادل اطلاعات بین گره ها بر اساس پروتکل AMQP, HTTP, CoAP, MQTT می پردازیم. جهت پیاده سازی پروتکل ها با حداقل جزئیات، ما باید در مورد چگونگی کار هر یک از پروتکل ها بدانیم. در شکل ۲، معماری شبکه را مشاهده می کنید.



شکل ۲- معماری کلی سیستم

همانطور که در شکل ۲ نشان داده شده است گره ۱ بر پایه‌ی یک ماژول NodeMcu است. سرور مرکزی کامپیوتر شخصی است که به عنوان سرور مرکزی استفاده می‌شود. گره ۲ یک ماشین مجازی بر روی سرور مرکزی است. شبکه ارتباطی این معماری wifi است. در این تمرین قصد پیاده سازی سناریو زیر را داریم:

گره ۱ در اتاقی قرار دارد و با استفاده از حسگر LDR میزان نور اتاق را به صورت عددی بین صفر تا صد ارسال می‌کند (این عملیات در تمرین اول بخش ۵ انجام شده است). گره ۲ با آگاه شدن از وضعیت نور اتاق، قادر به تنظیم کردن LED on-board بر روی گره ۱ خواهد بود، و مقدار ON را در صورت مشاهده میزان نور کمتر از ۵۰ جهت روشن شدن LED ارسال می‌کند و مقدار OFF را در صورت مشاهده میزان نور بیشتر از ۵۰ برای خاموش شدن LED ارسال می‌کند. و به این صورت نور اتاق به خوبی مدیریت می‌شود.

### بخش اول راه اندازی پروتکل MQTT

معماری ساختار MQTT شامل یک کارگزار (broker) است که با یک یا چند دستگاه ارتباط برقرار می‌کند. پایه ارتباطات بر اساس پیام‌ها و تاپیک‌هایی (publish/subscribe) است که به یک گره ارسال می‌شود. در این بین دیگر گره‌ها نیز می‌توانند آن پیام را دریافت کنند. سرور مرکزی مسئول مدیریت شبکه و انتقال پیام‌ها است.

1. بر روی سیستم شخصی کارگزار MQTT را نصب کنید. و ای پی و پورت مورد استفاده در MQTT server را نشان دهید. در این خصوص می‌توانید از broker متن باز مانند Mosquitto، EMQ، Mosca استفاده کنید. پیشنهاد می‌شود از MQTT broker mosquitto استفاده شود.

2. کتابخانه‌های مورد نیاز برای MQTT client را نصب کنید و سناریو گفته شده را اجرا کنید. این پروتکل تقریباً برای اکثر زبان‌های برنامه نویسی نسخه کلاینت دارد. به همین جهت پیشنهاد می‌شود برای گره ۱ کتابخانه مناسب با NodeMCU استفاده شود و گره ۲ را با زبان python پیاده سازی کنید.

3. پارامترهایی که در بسته ارسالی از کلاینت به سرور و برعکس وجود دارد را توضیح دهید.

### بخش دوم راه اندازی پروتکل CoAP

پروتکل CoAP بر مبنای درخواست/پاسخ (request/response) کار می‌کند و از کامنت‌های Get, Post, Put و Delete برای دریافت اطلاعات استفاده می‌کند.

1. بر روی سیستم شخصی CoAP server را نصب کنید. و ای پی و پورت مورد استفاده در CoAP server را نشان دهید. در این خصوص می‌توانید از هر CoAP server متن باز استفاده شود.

2. کتابخانه های مورد نیاز برای CoAP client را نصب کنید و سناریو گفته شده را اجرا کنید. این پروتکل تقریباً برای اکثر زبان های برنامه نویسی نسخه کلاینت دارد. در این خصوص می توانید از هر کتابخانه برای گره ۱ و گره ۲ استفاده کنید.

3. پروتکل CoAP و MQTT را از لحاظ معماری، مصرف انرژی، امنیت، کیفیت سرویس، سباز بسته ارسالی، با یکدیگر مقایسه کنید.

**بخش سوم راه اندازی پروتکل HTTP** این بخش به عنوان جبرانی برای افرادی که دو بخش آخر تمرین اول را انجام ندادند و نمره اضافه برای بقیه در نظر گرفته شده است)

پروتکل HTTP از دست تکانی (Handshaking) به منظور ارسال و دریافت اطلاعات استفاده می کند. در این روش برای شروع و پایان عملیات تبادل اطلاعات بین سرور و کلاینت، چندین درخواست و پاسخ رد و بدل می شود. لازم به ذکر است که در این فرآیند از پروتکل های TCP/IP به منظور افزایش امنیت و تضمین ارسال داده ها استفاده می شود.

1. بر روی سیستم شخصی HTTP server را نصب کنید. و ای پی و پورت مورد استفاده در HTTP server را نشان دهید. برای HTTP server پیشنهاد می شود از Nginx استفاده شود.

2. کتابخانه های مورد نیاز برای HTTP client را نصب کنید و سناریو گفته شده را اجرا کنید.

3. دلایل جایگزین شدن پروتکل MQTT و CoAP به جای HTTP در بسیاری از کاربردهای اینترنت اشیا را توضیح دهید.

**بخش چهارم راه اندازی پروتکل AMQP** (انجام این بخش نمره اضافه دارد)

پروتکل AMQP یا Advanced Message Queuing Protocol یک استاندارد متن باز برای انتقال پیام های بین گره ها است. معماری ساختار AMQP شامل یک کارگزار (broker) است که با یک یا چند دستگاه ارتباط برقرار می کند. دو عملیات exchanges و queues برای انتقال اطلاعات از گره منتشرکننده پیام (publish) به دریافت کننده پیام (subscribe) در این پروتکل انجام می شود.

1. بر روی سیستم شخصی کارگزار AMQP (broker) را نصب کنید. و ای پی و پورت مورد استفاده در AMQP broker را نشان دهید. در این خصوص می توانید از هر AMQP broker متن باز استفاده شود.

2. کتابخانه های مورد نیاز برای AMQP client را نصب کنید و سناریو گفته شده را اجرا کنید.

## نحوه تحویل تمرین

1. تمامی بخش های تمرین در قالب فایل ویدئویی جداگانه حداکثر ۲ دقیقه ای توضیح داده شود. برای بخش هایی که نیاز به توضیح دارد در قالب یک اسلاید توضیح داده شود.
2. ویدئو باید مشخص شده باشد که متعلق به شما است. به همین جهت قبل از توضیح مراحل انجام کار یک فایل بر روی سیستم خود نشان دهید که مشخص کند این ویدئو توسط شما ضبط شده است.
3. تمرین در قالب یک فایل zip تحویل داده شود و باید شامل فیلم ویدئویی بخش ها، کدها و اسلاید بخش های توضیحی باشد.
4. فیلم های ویدئویی هر بخش را به صورت زیر نام گذاری نمایید.
  - بخش اول MQTT (01-MQTT Broker, 02-MQTT scenario, 03-MQTT packet)
  - بخش دوم CoAP (01-CoAP server, 02- CoAP scenario, 03- CoAP Comparison)
  - بخش سوم HTTP (01- HTTP server, 02- HTTP scenario, 03- HTTP Comparison)
  - بخش چهارم AMQP (01- AMQP server, 02- AMQP scenario)
5. تمامی ویدئوهای ضبط شده باید قابل پخش با آخرین نسخه نرم افزار KMPlayer باشد.
6. مهلت تحویل تمرین، روز دوشنبه ۹ تیرماه خواهد بود.
7. به ازای هرروز تأخیر 5٪ جریمه در نظر گرفته خواهد شد.

موفق و مؤید باشید