



Department of  
Computer Engineering

به نام خدا



Amirkabir University of Technology  
(Tehran Polytechnic)

دانشگاه صنعتی امیرکبیر  
دانشکده مهندسی کامپیوتر  
مبانی اینترنت اشیا

گزارش بخش‌های تئوری تمرین سری چهارم

نام و نام خانوادگی	محمد توکلی
شماره دانشجویی	۹۷۳۱۰۱۴

**فهرست گزارش سوالات (لطفاً پس از تکمیل گزارش، این فهرست را به روز کنید.)**

- سوال ۱ - عنوان سوال ..... ۳
- سوال ۲ - عنوان سوال ..... ۴
- سوال ۳ - عنوان سوال ..... ۵
- سوال ۴ - عنوان سوال ..... ۶

## سوال ۱ - عنوان سوال

پروتکل COAP از UDP و HTTP از TCP و MQTT نیز از TCP در لایه transport استفاده میکنند.

پروتکل COAP در لایه نتورک از 6LoWPAN و پروتکل های HTTP و MQTT از لایه IP استفاده میکنند. پروتکل COAP و MQTT از multicast پشتیبانی میکنند درحالی که پروتکل HTTP اینطور نیست.

پروتکل COAP از هر دو مدل client-server و pub-sub استفاده میکند درحالی که HTTP فقط از client-server و MQTT از sub-pub استفاده میکند.

HTTP نیاز به ارتباطات سنکرون دارد درحالی که COAP و MQTT نیازی به این مکانیزم ندارند.

HTTP نسبت به MQTT و COAP دارای overhead بیشتری است و پروتکل سنگین تری است.

پروتکل های COAP و MQTT برای نود های محدود طراحی شده اند درحالی که HTTP برای دیوایس هایی طراحی شده است که از نظر منابع محدودیتی ندارند.

اندازه پیام ها در MQTT و COAP کم ولی در HTTP زیاد است.

پروتکل MQTT از SSL/TLS و پروتکل COAP از DTLS برای امنیت داده ها استفاده میکنند درحالی که HTTP از مکانیزمی استفاده نمیکند.

مدل ارتباطی COAP یک به یک درحالی که مدل ارتباطی MQTT چند به چند است. پروتکل COAP نسبت به MQTT برای نودهای محدود و در اینترنت اشیاء مناسب تر است زیرا از UDP به جای TCP استفاده میکند که سبک تر است و overhead کمتری دارد.

همچنین پروتکل COAP به اندازه MQTT قابلیت اطمینان ندارد زیرا MQTT از TCP استفاده میکند که دارای مکانیزم های data retransmission و ارسال ack و timeout و ... میباشد در حالی که COAP از UDP استفاده میکند و acknowledgment برای هر بسته ندارد اما میتوانیم با ست کردن فلگی در هدر بسته COAP آن را برای این پروتکل فعال کنیم تا مانند TCP بعد از ارسال هر بسته ack را نیز ارسال کند.

## سوال ۲ - عنوان سوال

یکی از مشکلات بزرگ COAP قابلیت پایین آن می باشد به دلیل اینکه این پروتکل از UDP استفاده میکند پس مکانیزم های ارسال مجدد بسته در صورت نرسیدن بسته یعنی ارسال های ACK بعد از ارسال هر بسته، timeout و ... را ندارد. راه حل این مشکل این است که یک فلگ را در هر بسته های COAP ست کنیم که این فلگ نشان میدهد ما میخواهیم برای آن بسته هایی که این فلگ ست شده است ACK ارسال شود.

از مشکلات بزرگ COAP میتوان به IP spoofing اشاره کرد. بدین صورت که شخص حمله کنند آدرس IP خود را با آدرس IP مقصد جاگذاری میکند و خود را به جای آن جا میزند. راه حل آن استفاده از TLS است که درست است مشکل ip spoofing را برطرف میکند اما باعث سنگین شدن پروتکل میشود.

یکی دیگر از مشکلات COAP، packet amplification میباشد. بدینصورت که attacker یک بسته برای کلاینت ارسال میکند و کلاینت با یک پکت بزرگتر جواب میدهد که سائز آن چندین برابر پکت دریافت شده است. برای برطرف کردن این مشکل نیز از TLS استفاده میکنیم.

از مشکلات MQTT میتوان به single point of failure بودن آن اشاره کرد. زیرا این پروتکل از مدل pub-sub برای تبادل پیام استفاده میکند و دو قسمت broker و client دارد. کلاینت ها پیام خود را برای broker میفرستند و broker نیز پیام هارا برای client های دیگر. بنابراین broker میتواند bottleneck باشد. برای رفع این مشکل میتوانیم به جای یک broker که bottleneck ما میشود از چندین broker استفاده بکنیم که در صورتی که یکی از broker ها از کار بیفتد بقیه بتوانند کار آن را انجام دهند و سیستم تبادل پیام از کار نیفتد.

### سوال ۳ - عنوان سوال

در اینترنت اشیاء برای برقراری ارتباط بین دستگاه های مختلف لازم داریم که دستگاه هارا بشناسیم و دستگاه های پایانی به اصطلاح end-device ها از هم متمایز باشند. زیرا دستگاه های بسیار زیادی قرار است بهم وصل شوند و با platform ها و application در ارتباط باشند. پس باید تمایزی بین شیء های مختلف باشد.

یکی از راه هایی که برای تمایز اشیاء استفاده میشود؛ استفاده از DevEUI میباشد. بدینصورت که یک شناسه یکتا هر شرکت سازنده device ها برای آن device در نظر میگیرند که از بقیه اشیاء در اینترنت اشیاء متمایز باشند.

DevEUI یک شناسه ۶۴ بیتی است که شرکت سازنده سنسور ها و عملگر ها به آن شیء میدهند که این شناسه شامل، شناسه یکتا شرکت سازنده آن، ورژن سنسور و یا عملگر مورد استفاده و همچنین شناسه سنسور و عملگر تولیدی میباشد که شناسه سنسور ها و عملگر ها نیز از هم متمایز هستند. این مورد باعث میشود حتی اشیاء ساخته شده توسط شرکت های مختلف نیز از یکدیگر متمایز باشند زیرا در این شناسه، شناسه خود شرکت سازنده را نیز قرار میدهیم که باعث میشود شرکت های سازنده نیز از هم متمایز باشند. زیرا شناسه شرکت ها باید ثبت جهانی شوند و نمیشود که شناسه دو شرکت باهم برابر باشند. همچنین درصورت ثابت بودن شرکت سازنده آیدی های اشیاء تولید شده توسط آن شرکت نیز متمایز است که باعث میشود اشیاء ساخته شده توسط یک شرکت نیز متمایز باشد. همچنین ورژن نیز باعث میشود اشیاء یکسان ساخته شده توسط یک شرکت یکسان نیز با ورژن های مختلف از هم تمایز پیدا کنند. بدین ترتیب اشیاء مختلف که توسط شرکت های مختلف ساخته شده اند در اینترنت اشیاء از هم متمایز میشوند و میتوانند باهم ارتباط برقرار کنند.

## سوال ۴ - عنوان سوال

در اینترنت اشیاء نیازمند این هستیم که بین اشیاء مختلف اطلاعات تبادل کنیم و اطلاعات سنسور ها را برای عملگر ها بفرستیم و یا آنها را به سمت platform و یا storage بفرستیم. چون اطلاعات مختلفی جابجا میشود پس نیازمند مدل های اطلاعاتی (data model) هستیم. یک انتخاب این است که برای تمام اشیاء بخواهیم مدل های اطلاعات متفاوت در نظر بگیریم. این روش یک اشکال بزرگ دارد و آن این است که در این شرایط سربار ها برای همدل کردن حالت ها و استاندارد های مختلف برای دستگاه ها در اینترنت اشیاء افزایش میابد و حجم محاسبات زیاد میشود و از آنجا که اشیاء در اینترنت اشیاء همگی محدود هستند یعنی از نظر پردازشی، حافظه، شبکه و ... محدود هستند در نتیجه این راه گزینه مناسبی نیست زیرا استاندارد های مختلف زیادی وجود خواهد داشت و در نظر گرفتن تمامی آنها حجم محاسباتی ما را افزایش میدهد. پس در عمل و در دنیای واقعی باید از یک دیتامدل یکسان استفاده کنیم.

یکی از این راه حل ها استفاده از SenML میباشد. SenML یک مدل استاندارد و کدگذاری XML برای توصیف کردن هر پردازشی شامل اندازه گیری مقادیر سنسور ها میشود. از SenML برای توصیف طیف گسترده ای از سنسور ها استفاده میشود و برای سنسور های ساکن و متحرک و نزدیک و دوربر استفاده میشود. داده در SenML یک آرایه تکی شامل رکورد های مختلف SenML میباشد. که شامل شناسه های سنسور ها، زمانی که اندازه گیری سنسور انجام شد، واحد اندازه گیری و مقدار فعلی سنسور میشود. به عنوان مثال داده زیر مقدار اندازه گیری شده توسط یک سنسور است که در فرمت JSON در SenML کدگذاری شده است:

```
[
  { "n": "urn:dev:ow:10e2073a01080063", "u": "Cel", "v": 23.1 }
]
```