

بسم الله الرحمن الرحيم



دانشگاه صنعتی امیرکبیر  
( پلی تکنیک تهران )

پاسخنامه تکلیف ۴ درس اینترنت اشیا

دکتر خرسندی

محمد مهدی منتظر

۹۷۳۱۱۲۰

سید امیرمهدی میرشریفی

۹۸۳۱۱۰۵

## بخش تئوری:

### سوال 1: (امتیاز: ۵)

سه مورد از دلایلی که HTTP برای small devices مناسب نیست را نام ببرید و به طور مختصر توضیح دهید.

پاسخ:

از انجایی که HTTP بیشتر از پروتکل TCP استفاده میکند و این پروتکل نیاز به یک بافر دارد همچنین بسته ها در این پروتکل مستلزم یک شمارنده هستند تا TCP در صورت شک در ارسال مطمئن داده ها آن ها را مجدداً ارسال کند ، دیوایس هایی که بخواهند در لایه انتقال از این پروتکل استفاده کنند نیاز به پیاده سازی پیچیده تری دارند و همچنین نیاز به منابع بیشتری نسبت به نود هایی هستند که این فرآیند را از طریق پروتکل UDP پیاده سازی میکنند دارند. از این رو دستگاه های کوچک با استفاده از پروتکل UDP عملیات انتقال را انجام میدهند که موجب پیاده سازی راحت تر و استفاده بهینه تر از منابع است.

- HTTP is based on TCP, the data stream-based transmission method in the TCP/IP stack.
    - Among other things, TCP requires that data packets are equipped with counters in order to be able to correct the order of incoming packets in case of doubt.
    - TCP devices need buffers to store and manage packets between them.
  - Many small devices therefore use UDP, which delivers datagrams individually and does not guarantee delivery or the actual order.
    - This makes the implementation leaner and more resource-efficient.
- همچنین خود پروتکل Http نیز به ۴ دلیل مناسب کاربرد های IoT نیست که در ادامه به آن ها می پردازیم:

(۱) پروتکل Http یک پروتکلی با هدر های زیاد است که این پیچیدگی و سنگینی برای فناوری های IoT سنگین است

- HTTP is a heavy weight protocol with many headers and rules
  - HTTP is a chatty protocol from the point of view of resource restriction.
    - For example, meta information such as the accepted formats or the desired language are transmitted to the server in plain text form.
  - The header of an HTTP request can already be large, and the header of an HTTP response can easily contain several hundred bytes.
  - For the query of a web page with several megabytes of size this is not bad, but for the query of a temperature value in the range of approx. 5 bytes it is significant whether the transmission protocol is slim or rather broad.

(۲) پروتکل Http نیاز به همگام سازی میان کلاینت و سرور دارد که در کاربرد های اینترنت اشیا ما تعداد بسیار زیادی سنسور و دستگاه های متعدد داریم که همگام سازی میان این حجم از شی بهینه نخواهد بود.

#### – HTTP is a synchronous protocol.

- The client waits for the server to respond.
- That is a requirement for web browsers, but it comes at the cost of poor scalability.
- In the world of IoT, the large number of devices and most likely an unreliable / high latency network have made synchronous communication problematic.
- An asynchronous messaging protocol is much more suitable for IoT applications. The sensors can send in readings, and let the network figure out the optimal path and timing for delivery to its destination devices and services.

۳) در پروتکل Http شروع کننده درخواست کلاینت و پاسخ دهنده سرور است و اغلب کلاینت کامند های مختلف را سمت سرور ارسال میکند و پاسخ میگیرد ، در حالی که کاربرد های اینترنت اشیا هر دو طرف نیاز به تبادل دستوراتی و گرفتن پاسخ های متناسب با آن دستور ها هستند.

– HTTP is one-way.

- The client must initiate the connection.
- In an IoT application, the devices or sensors are typically clients, which means that they cannot passively receive commands from the network.

۴) در پروتکل Http امکان برادکست و انتشار یک پیام برای دیوایس های متعدد وجود ندارد و برای ارسال پیام حتما باید ارتباط با مقصد برقرار شود

– HTTP is a 1-1 protocol.

- The client makes a request, and the server responds.
- It is difficult and expensive to broadcast a message to all devices on the network, which is a common use case in IoT applications.

بنابر دلایلی که مطرح شد پروتکل HTTP نه تنها برای دستگاه های محدود و کوچک مناسب نیست بلکه برای کاربرد های IoT نیز مناسب نمی باشد.

## سوال 2: (امتیاز: ۵)

با توجه به پروتکل CoAP به سوالات زیر پاسخ دهید:

- الف) توضیح دهید این پروتکل از چه جهت هایی مشابه HTTP است و تفاوت این دو عمدتا در چیست؟
- ب) یک CoAP Message شامل چه بخش هایی است؟ و نقش Payload را در آن توضیح دهید.
- ج) چه زمانی CoAP Message از نوع Confirmable است؟

پاسخ:

الف) در واقع CoAP نسخه سبک شده HTTP برای کاربرد های IoT است . از شباهت های متد های مورد استفاده ، نوع پیام های دریافتی اشاره کرد هر چند که تنوع آن ها کمتر است زیرا مثل HTTP سنگین نیستند . همچنین CoAP همانند HTTP از URI برای دسترسی به منابع استفاده میکند.

- A uniform resource identifier (URI) localized on the server identifies this resource.
- CoAP Messages semantics
  1. Get
  2. Post
  3. Put
  4. Delete

از تفاوت های CoAP با HTTP میتوان به این اشاره کرد که:

(۱) HTTP از TCP و CoAP از UDP استفاده میکند که دلایل آن در سوال یک نیز بیان شد از این رو عملاً لایه انتقال توان رساندن قابل اعتماد پیام ها را ندارد و اگر بخواهیم این قابلیت را ایجاد کنیم می بایست در لایه اپلیکیشن آن را پیاده سازی کنیم که برای این کار ۴ نوع پیام ایجاد میکنیم که دو نوع اول مشخص میکنند آیا نیاز به ACK هست یا خیر .

(۲) تاخیر کمتر CoAP نسبت به HTTP

- CoAP is upgraded version of HTTP.
  - It is designed for resource constrained applications such as IoT/WSN/M2M etc.
- CoAP is based on UDP.
  - It uses ACK messages so that it will become reliable like TCP.
- CoAP has low latency and consumes lesser power compare to HTTP.

(۳) در HTTP همگام سازی وجود دارد اما در CoAP نیازی به همگام سازی نیست.

- The asynchronicity of message exchange via UDP also gives CoAP a device the possibility of responding to a request in a resource-saving and time-delayed manner, as well as of transmitting a response in several small (partly “smallest”) blocks.

(۴) نوع خروجی در HTTP متن است و هدر های سنگین اما در CoAP خروجی به صورت باینری است.

- Differences of CoAP with HTTP result mainly in the amount of data used.
  - In contrast to the plain text format of HTTP, especially the header format, CoAP code words are packed in bytes or individual bits of a binary representation.

همچنین در جدول زیر میتوانید تفاوت های دیگری مثل معماری یا لایه شبکه و غیره را مشاهده کنید:

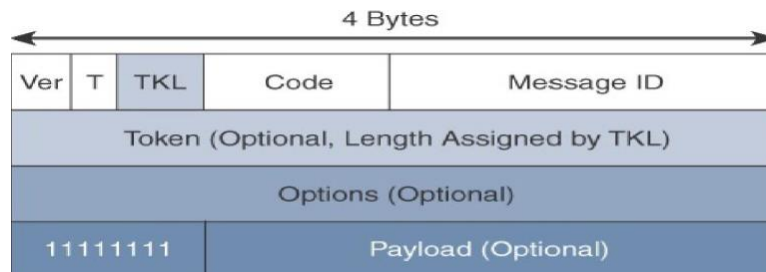
Feature	CoAP	HTTP
Protocol	It uses UDP.	It uses TCP.
Network layer	It uses IPv6 along with 6LoWPAN.	It uses IP layer.
Multicast support	It supports.	It does not support.
Architecture model	CoAP uses both client-Server & Publish-Subscribe models.	HTTP uses client and server architecture.
Synchronous communication	CoAP does not need this.	HTTP needs this.
Overhead	Less overhead and it is simple.	More overhead compare to CoAP and it is complex.
Application	Designed for resource constrained networking devices such as WSN/IoT/M2M.	Designed for internet devices where there is no issue of any resources.

ب) CoAP message شامل سه بخش است :

بخش اول بخش هدر پیام است که ۴ بایت است و شامل فیلدهایی مثل ورژن و نوع پیام و .... است.

بخش دوم بخش توکن است که طول آن متغیر است ( صفر تا هشت بایت ) .

بخش سوم بخش فیلدهای اختیاری و payload است که حاوی داده پیام است .



ج) همانطور که پیش تر بیان شد با توجه به این که CoAP از UDP استفاده میکند از این رو امکان ارسال مطمئن داده ها را ندارد بنابراین ما ۴ نوع پیام تعریف میکنیم که در واقع این انواع پیاده سازی ای برای انتقال مطمئن پیام هستند. پیام هایی که از نوع confirmable هستند به این معنی هستند که ارسال مطمئن آنها برای ما اهمیت دارد . از این رو است که بعد از ارسال این نوع پیام ها ، یک نوع پیام از جنس ACK دریافت خواهیم کرد . باز هم تاکید میشود که این عملکرد پیاده سازی ای در بخش اپلیکیشن است .

### سوال 3: (امتیاز: ۵)

در مورد پروتکل MQTT به سوالات زیر پاسخ دهید:

الف) مفهوم QoS در MQTT Message به چه معناست؟ حالت های مختلف آن را توضیح دهید.

ب) نقش broker در یک سرور MQTT چیست؟ و چه انواعی دارد؟

پاسخ:

الف) QoS به معنی کیفیت سرویس است که در سه سطح ارائه میشود و همچنین فیلد آن در پیام بیانگر این است که کدام یکی از سطوح در حال استفاده است :

سطح Q0 : این سطح از کیفیت بیانگر این است که پکت ارسالی حداکثر یک بار به واسطه گیرنده دریافت میشود ، یعنی یا یک بار دریافت میشود یا دریافت نمی شود . در این سطح هیچ پاسخی از طرف گیرنده دریافت نخواهد شد

سطح Q1: این سطح از کیفیت بیانگر این است که حداقل یکبار پکت ارسالی توسط سرور و دنبال کننده ا دریافت میشود و در صورت عدم دریافت ACK یا همان PUBACK پیغام دوباره ارسال خواهد شد. این سطح تضمین میکند پیام حداقل یکبار دریافت میشود.

سطح Q2 : این سطح از کیفیت بیانگر این است که پیغام دقیقاً یکبار توسط دنبال کننده ها دریافت میشود ، نه مانند سطح صفر که حداکثر یک بار نه مانند سطح یک که حداقل یک بار ، دقیقاً یکبار.

ب) broker یک واسطه بین انتشار دهنده پیام و دنبال کننده های آن انتشار دهنده است بدین صورت که انتشار دهنده پیام را برای broker ارسال میکند و سپس broker آن را برای تمامی دنبال کنندگان ارسال میکند.

The broker is a server that receives all messages from the clients and then routes those messages to relevant destination clients.

A client is anything that can interact with the broker to send and receive messages.

از انواع آن میتوان به

Open source like verneMQ

Commercial like EMQ

Cloud(managed) like Azure IoT Hub

اشاره کرد.