به نام خدا

دانشگاه صنعتی امیرکبیر

دانشکده مهندسی کامپیوتر

مبانی اینترنت اشیا

گزارش بخش‌های تئوری تمرین سری چهارم

نام و نام خانوادگی	دانیال حمدی
شماره دانشجویی	۹۷۳۱۱۱۱

فهرست گزارش سوالات

سوال ۱ – مقایسه‌ی پروتکل‌های MQTT, CoAP, HTTP

	MQTT	CoAP	HTTP
معماری کلی	PubSub	PubSub + ReqResp	ReqResp
مسیر داده	یک به یک/چند	می‌تواند یک‌به‌یک یا یک‌به‌چند باشد.	یک به یک
نیاز به هم‌زمانی فرستنده و گیرنده (سنکرون یا آسنکرون)	ندارد.	می‌تواند داشته باشد. (در حالت ReqResp)	دارد.
پروتکل زیرین مورد استفاده	پروتکل TCP	پروتکل UDP	عموماً پروتکل TCP (هر چند که از UDP هم می‌تواند استفاده کند.)
قابلیت اطمینان	دارد.	نسبت به MQTT از قابلیت اطمینان کم‌تری برخوردار است. چون از UDP استفاده می‌کند. اما در مقابل سریع‌تر است.	دارد.
امنیت	از TLS/SSL استفاده می‌کند.	از پروتکل DTLS استفاده می‌کند.	نسخه‌ی امن شده‌ی آن، نسخه‌ی HTTPS می‌باشد.
فرمت انتقال داده	باینری، ۲ بایت.	باینری.	متنی، که منجر به هدرهایی مفصل‌تر و سنگین‌تر می‌شود.

سوال ۲ – مشکلات پروتکل‌های MQTT و CoAP و راه حل‌های آن

مشکلات و راه‌حل‌های MQTT

- امنیت: امنیت به طور پیش‌فرض در MQTT تعبیه نشده است. برای اضافه کردن امنیت، باید خودمون باید یک لایه روی آن سوار کنیم.
- دستگاه‌های کم‌توان Constrained Node امکان اجرای MQTT را ندارند. پس باید با یک پروتکل سبک‌تر به یک دستگاه پرتوان‌تر متصل شده و آن دستگاه به MQTT متصل شود.
- این پروتکل امکان ایجاد Single Point of Failiure را دارد. مؤلفه‌ی مرکزی ارتباطات مختلف، Broker می‌باشد. در صورتی که تنها یک Broker داشته باشیم، با بروز مشکل در آن، تمامی ارتباطات تأثیر می‌پذیرند. برای رفع چنین مشکلی باید به سمت معماری‌هایی با چند Broker برویم.
- این پروتکل برای کاربردهایی مانند Streaming مناسب نیست.

منبع

مشکلات و راه‌حل‌های CoAP

- امنیت: این پروتکل به طور پیش‌فرض رمزنگاری شده نیست. برای چنین منظوری، باید خود راه حل‌های رمزنگاری پیام‌ها را پیاده کنیم.
- امنیت: آسیب‌پذیری‌های پروتکل زیرین CoAP یعنی UDP، سطح امنیت آن را پایین می‌آورند. پروتکل UDP در برابر جعل IP یا مواردی از این دست آسیب‌پذیر است.
- اطمینان‌پذیری (Reliability) کامل نداریم. این مسئله هم به دلیل استفاده از UDP است، که البته مزایایی مانند سادگی و سرعت را به ارمغان می‌آورد. برای بالا بردن قابلیت اطمینان خود CoAP در پیام‌هایش بیت‌هایی برای ACK یا NACK در نظر گرفته است.

سوال ۳ – شناسه‌دهی یکتا به End Deviceها

نیاز صنعت IoT به شناسه‌دهی یکتا به End Deviceها امری واضح است. در کاربردهای مختلف، پس از دریافت داده‌های مختلف از Thingها (در واقع حسگرها) و یا پیش از ارسال دستور به عملگرها، نیاز داریم دستگاه هدف خود را به طور یکتا مشخص کنیم.

تعداد بسیار بالای دستگاه‌های End Device در IoT، نیاز به شناسه‌های واحد و هماهنگ را عمیق‌تر می‌کند. (Scalability) در صورتی که شناسه‌های هر گروه دستگاه با دیگری متفاوت باشد، همکاری بین دستگاه‌های مختلف (Interoperability) امری دشوار و حتی ناممکن خواهد بود.

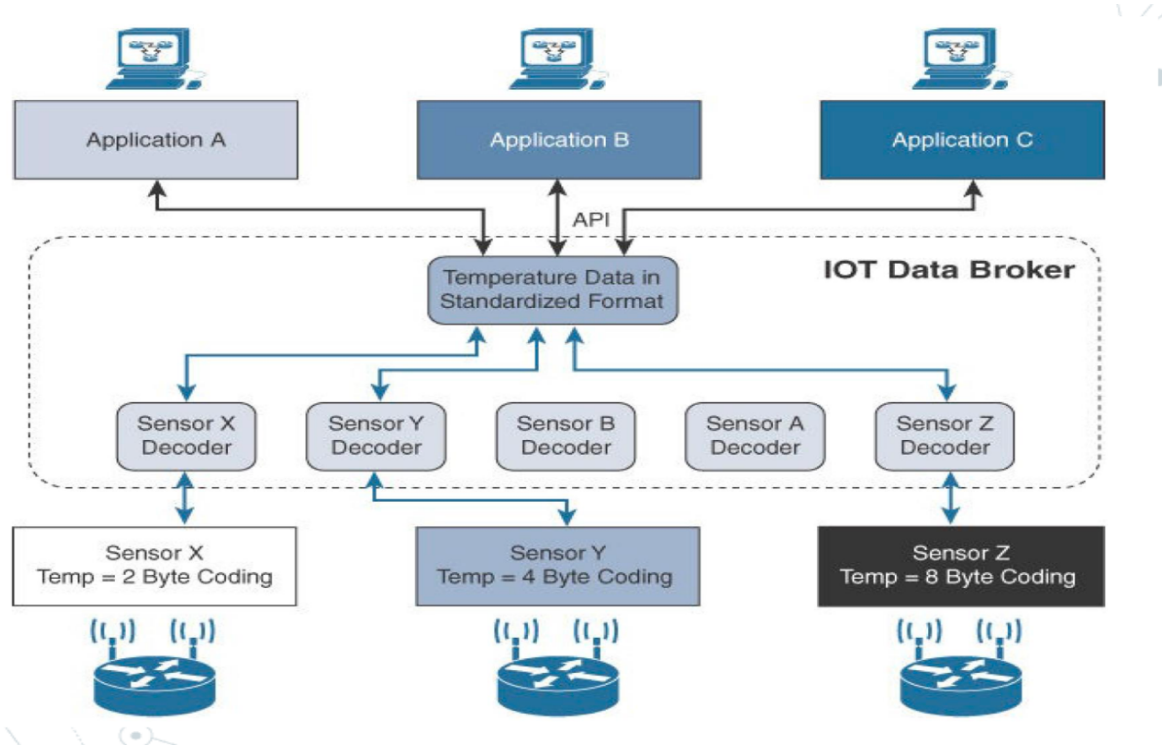
در LoRa برای حل این مشکل، یک شناسه‌ی ۶۴ بیتی در نظر گرفته می‌شود. این رشته‌ی بیتی هر چند توان در خود نگاه داشتن ۲ به توان ۶۴ حالت مختلف از رشته‌ها را دارد، اما همچنان محدود است. پس به هر سازنده‌ی دستگاه‌های IoT تنها بخش خاصی از بازه‌ی مجاز داده می‌شود. رشته‌های این بازه‌ی اختصاص یافته× همگی در بخشی از رشته که مختص سازنده است، یکسان می‌باشند. بخش‌های دیگر برای مشخص کردن نوع و ویژگی‌های دستگاه می‌باشد.
منبع ۱

راه حل‌های مشابه دیگری نیز برای اختصاص شناسه در نظر گرفته شده است. مثلاً می‌توان فضای کاربرد End Deviceها را به بخش‌های مختلف شکست (Namespaceهای مختلف) و در هر Namespace برای End Deviceها یک Name یکتا در نظر گرفت. بدین صورت دیگر نیازی به یکتا بودن Name در تمام Namespaceها نیست و Namespaceهای مختلف از یکدیگر جدا می‌شوند.
منبع ۲

سوال ۴ – مدل اطلاعاتی SenML

روش SenML برای هماهنگ‌سازی مدل اطلاعاتی Thingهای مختلف

برای ارتباط ساده‌تر بین دستگاه‌های مختلف IoT (مفهوم Interoperability) نیاز به یک زبان مشترک انتقال داده داریم.
(Standardization). دستگاه‌های ناهم‌خوان، منجر به صرف انرژی بیش‌تر برای ترجمه و یکپارچه‌سازی فرمت داده‌های مختلف می‌شود.
مثلاً برای مثال گفته شده‌ی اندازه‌گیری رطوبت خاک، اگر هر حسگر به شیوه‌ی خاص خودش داده‌های اندازه‌گیری شده را اندازه‌گیری کند، آن‌گاه نیاز به یک مؤلفه‌ی اضافه به نام Data Broker داریم که داده‌ها با فرمت مختلف را جمع‌آوری کرده و آن‌ها رو به فرمت یکسان در می‌آورد.



پرواضح است که اجتناب از این ناهم‌خوانی مدل اطلاعاتی، معماری سیستم و هزینه‌های آن را کاهش می‌دهد.
مدل SenML یا Sensor Measurements List که در RFC ۸۴۲۸ مستند شده، برای رفع همین مشکل، و معرفی مدل داده‌ی یکپارچه ایجاد شده است. مثالی از این ارسال داده در این مدل را در تصویر زیر می‌بینیم.
هر داده در فرمت این مدل، می‌تواند کلیدهای زیر را داشته باشد. (در بازنمایی JSON)

Name	Label	CBOR Label
Base Version	bver	-1
Base Name	bn	-2
Base Time	bt	-3
Base Unit	bu	-4
Base Value	bv	-5
Base Sum	bs	-6
Name	n	0
Unit	u	1
Value	v	2
String Value	vs	3
Boolean Value	vb	4
Sum	s	5
Time	t	6
Update Time	ut	7
Data Value	vd	8

مقادیر Base X (مثلاً Base Name یا Base Version) میزانی ثابت هستند که باید به مقدار X افزوده شوند.
مثلاً در تصویر زیر، نام کامل حاصل ترکیب bn (همان Base Name) با n (همان Name) می‌باشد.
از داده‌ی زیر در میابیم برای دستگاه urn:dev:ow:۱۰e۲۰۷۳a۰۱۰۸۰۰۶۳ میزان ولتاژ برابر ۱۲۰.۱ و میزان جریان برابر ۱.۲ آمپر می‌باشد.
کلید U واحد عدد گزارش شده در V را مشخص می‌کند. مثلاً مقادیر مثال زیر در دستگاه SI گزارش شده‌اند.

```
[
  {
    "bn": "urn:dev:ow:10e2073a01080063:",
    "n": "voltage",
    "u": "V",
    "v": 120.1
  },
  {
    "n": "current",
    "u": "A",
    "v": 1.2
  }
]
```