دانشگاه صنعتی اصفهان

دانشکده مهندسی برق و کامپیوتر

اینترنت اشیا

گزارش پروژه ی کارشناسی

محمد مهدی امینی

استاد

دکتر منشیی

۱۳۹۶

فهرست مطالب

فهرست تصاویر

فهرست جداول

فصل اول

MQTT

۱ـ۱ معرفی

MQTT یک پرتوکل استاندارد (ISO/IEC PRF 20922) ارتباطی است که به صورت اختصاصی برای کاربردهای اینترنت اشیا طراحی شده است. این پرتوکل به دلیل طراحی خاص خود در محیط های با پهنای باند محدود به خوبی کار می کند. از ویژگی های اصلی آن میتوان به سایز کوچک بسته ها و توان مصرفی پایین اشاره کرد. این پرتوکل همچنین برای محیط های Wireless با ارتباط های غیر قابل اتکا و دارای تاخیر متغیر مناسب است. به علاوه، این پرتوکل به شکلی طراحی شده تا پیاده سازی آن شامل پیچیدگی نباشد. کاربردهای این پرتوکل اساسا به اینترنت اشیا مربوط می شود. معروف ترین کابرد آن ارسال داده های تولید شده توسط سنسورها به یک سرور جمع آوری مقادیر است.

۱ـ۲ لایه ی انتقال

درلایه ی انتقال MQTT از پرتوکل TCP استفاده می کند. البته یک پرتوکل به نام MQTT-SN از این پرتوکل مشتق شده است که امکانات MQTT را در محیط های فاقد TCP/IP مثل ZigBee ارایه می کند. اگرچه در Quality of service, TCP وجود ندارد، اما در MQTT علی رغم سادگی اش، Quality of service به عنوان

یک ویژگی قابل اتکا در لایه ی Application طراحی شده است.

۱ـ۳ ساختار پیام

پیام های MQTT به صورت باینری منتقل می شوند. هر بسته ی پیام MQTT از سه section تشکیل شده. section اول هدر با طول ثابت است که ۲ byte طول دارد. section دوم هدر اختیاری است با طول متغییر. و section سوم Payload است که می تواند تا ۲۵۶ مگابایت داده را شامل شود. فقط section اول است که در تمام انواع مختلف پیام های این پرتوکل وجود دارد.

bit	7	6	5	4	3	2	1	0
byte 1	Message Type				DUP flag	QoS level		RETAIN
byte 2	Remaining Length							

شکل ۱ـ۱: شمای Header با طول ثابت

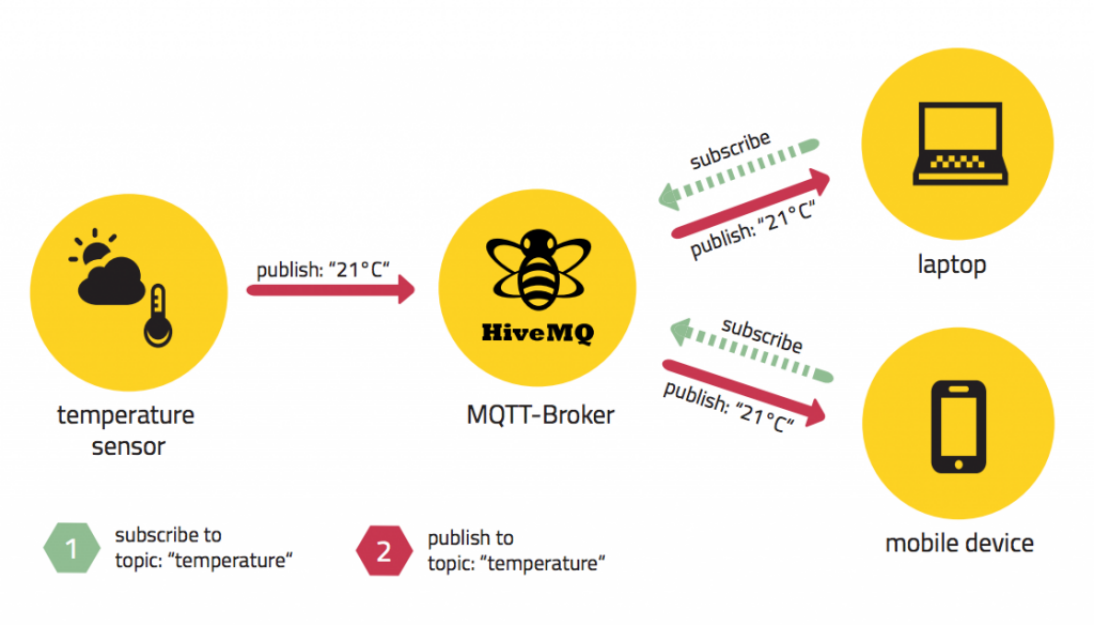
۱ـ۴ الگوی publish/subscribe

الگوی Publish/Subscribe یک روش ارتباطی جایگزین برای مدل سنتی کلاینت/سرور است. در روش کلاینت/سرور، کلاینت به صورت مستقیم با یک نقطه ی نهایی(Point End) که در واقع سرور است ارتباط بر قرار می کند. اما در مدل Pub/Sub طرفین ارتباط از وجود یکدیگر خبر ندارند. بعضی از موجودیت ها ارسال کننده ی پیام هستند(Publisher) و بعضی دیگر دریافت کننده (Subscriber) . یک موجودیت در این مدل، می تواند به صورت همزمان هم دریافت کننده و هم ارسال کننده ی پیام باشد. در معاری این مدل یک موجودیت سوم نیز وجود دارد که هم Publisher ها و هم Subscriber ها از وجود آن آگاه هستند و با آن ارتباط مستقیم بر قرار می کنند. این موجودیت Broker نام دارد. Broker پیام ها را از Publisher ها دریافت می کند. Subscriber ها هر زمان که به Broker متصل شدند، اگر پیامی برایشان در دسترس بود، پیام را دریافت می کنند. در واقع Broker وظیفه ی مدیریت و توزیع پیام ها را دارد و به نوعی شبیه یک مجموعه ی بزرگ از Buffer ها است.

در این الگو دریافت کننده های پیام نیاز ندارند نسبت به دانستن زمانی که اطلاعات جدیدی در دسترس قرار می گیرد نگران باشند. هر زمان که پیام (یا داده ی جدیدی) در دسترس باشد با کمترین میزان سربار از آن مطلع خواهند شد. در مدل سنتی کلاینت/سرور، کلاینت ها مجبور بودند برای بروز بودن و آگاهی از اخرین داده ها، هر چند وقت یکبار مثلا هر ده ثانیه یکبار به سرور یک درخواست ارسال کنند و درباره ی وجود پیام (یا داده ی جدید) سوال کنند که باعث به هدر رفتن منابع مختلف سرور و کلاینت ها در میزان بالا می شد.

با توجه به مستقل بودن Entity ها و مستقل بودن فرایند دریافت پیام از فرایند ارسال آن، امکان استفاده از

مزایای موازی سازی کاملا وجود دارد. با در نظر گرفتن event-driven بودن معماری می توان نتیجه گرفت که این مدل دارای قابلیت Scalability به میزان قابل توجهی است.



شکل ۱ـ۲: نمای کلی از مدل Pub/Sub در قالب یک مثال از MQTT

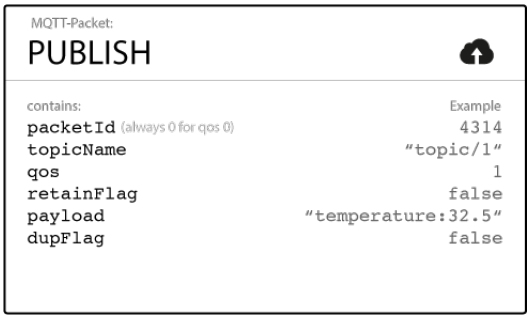
Client

در فضای این پرتوکل هر موجودیتی که توانایی Publish یا Subscribe از طریق اتصال به یک سرور مرکزی (Broker) را داشته باشد، کلاینت نامیده می شود. لازم به ذکر است که هر دو موجودیت کلاینت و سرور توانایی publish و subscribe کردن دارند. کلاینت ها به ۲ دسته ی Transient و Persistent تقسیم می شوند. کلاینت های Persistent جلسه ی ارتباطی خود با سرور مرکزی (Broker) را حفظ می کنند. اما ارتباط کلاینت های Transient توسط سرور مرکزی دنبال نمی شود(Session یا جلسه ای برای کلاینت نگه داری نمی شود.).

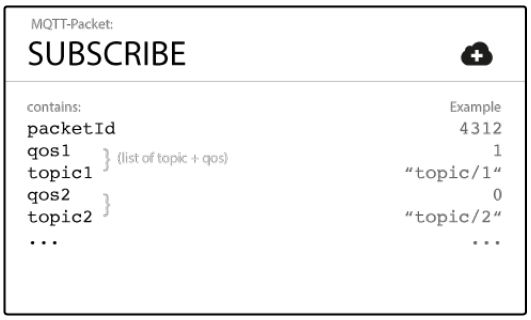
Topic

یک نقطه ی اتصال نهایی است که کلاینت ها به آن متصل می شوند. Topic در واقع به عنوان یک Hub مرکزی برای توزیع پیام ها عمل می کند. اگر چه باید Topic ها قبل از اتصال موجودیت ها به آنها ساخته شوند ولی در صورت عدم وجودشان، به محض اینکه یک موجودیت درخواست اتصال به آن را ارسال کند، بدون وقفه Topic مورد نظر ساخته می شود. Topic ها با آدرسشان که یک ساختار درختی مثل ساختار آدرس فایلها دارد، شناخته می شوند. برای مثال building۱/room۱/temperature آدرسی است که تاپیک مربوط به مدیریت پیام های سنسورهای دماسنج اتاق۱ در ساختمان Building را مشخص می کند. دریافت پیام از یک Topic

که در واقع Subscirbe کردن آن است به کمک آدرس آن انجام می شود. برای مثال building1/#/ آدرس Subscirbe کردن تمام Topic های زیر شاخه ی Building۱ است.



شکل ۱ـ۳: نمایشی متنی پیامی که منجر به Publish می شود در MQTT



شکل ۱ـ۴: نمایشی متنی پیامی که منجر به Subscribe می شود در MQTT

۱ـ۴ـ۱ جلسه ی MQTT

هر جلسه ی MQTT از چهار فاز connection, authentication, communication, termination تشکیل می شود. کلاینت ابتدا با ایجاد یک ارتباط با Broker شروع می کند(فاز Connection). ممکن است Broker جلسه ی جدیدی برای کلاینت ایجاد نکند و آخرین جلسه ای که کلاینت با آن به Broker متصل بوده را ادامه دهد. اتصال می تواند از طریق پورت های استاندارد ۱۸۸۳ برای ارتباط های عادی و ۸۸۸۳ برای ارتباط های SSL/TLS انجام گیرد. همچنین امکان تنظیم یک پورت دلخواه در بروکر برای ایجاد ارتباط وجود دارد.

سپس کلاینت، با بررسی گواهینمای سرور آن به اهراز هویت آن می پردازد. همچنین این امکان به صورت اختیاری برای کلاینت وجود دارد که گواهینامه ی خود را به سرور ارایه کند.(فاز Authentication).

پس از اهراز هویت سرور، امکان publish پیام به یک topic خاص یا subscribe کردن پیام های آن وجود دارد(فاز Communication).

سرور و کلاینت ها می توانند ارتباط TCP ای که با یکدیگر دارند را، خاتمه دهند(فاز Terminition).

۱ـ۵ امنیت

پرتوکل MQTT در حوزه ی امنیت با توجه به هدف طراحی آن که ساده و کم سربار بودن است، دارای ضعف های قابل بررسی ای است. با توجه به اینکه این پرتوکل معمولا برای ارسال داده های سنسورها و ارسال دستور به عملگرها استفاده می شود، اهراز هویت و محرمانگی از نیازهای مشترک اکثر کاربردهای آن است. امنیت به ۲ روش در این پرتوکل قابل دستیابی است.

۱ـ ارسال username و password به صورت clear-text . در این روش کلاینت اطلاعات کاربری خود را در پیام های MQTT به صورت شفاف قرار می دهد. این روش بسیار ابتدایی و به راحتی قابل دور زدن است. در نتیجه استفاده از آن پیشنهاد نمی شود.

۲ـ استفاده از پرتوکل SSL/TLS . در این روش کلاینت موظف است گواهینامه ی سرور را بررسی کند و ارایه ی گواهینما توسط کلاینت به صورت اختیاری امکان پذیر است. متاسفانه استفاده از این روش باعث اعمال پیچیدگی و سرباری می شود که با هدف ساده بودن MQTT در تضاد است.

کتاب‌نامه

[1] Washer, Peter. *Learning Internet of Things*. Packt, 2015.

[2] MQTT. http://mqtt.org.

[3] HIVEMQ, MQTT Essentials: Part 1 – Introducing MQTT. http://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt.

[4] HIVEMQ, MQTT Essentials: MQTT Essentials Part 2: Publish and Subscribe. http://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe.

[5] Janakiram MSV, Get to Know MQTT: The Messaging Protocol for the Internet of Things. https://thenewstack.io/mqtt-protocol-iot/, 2016.

[6] Margaret Rouse, MQTT (MQ Telemetry Transport). http://internetofthingsagenda.techtarget.com/definition/mqtt-mq-telemetry-transport, 2016.

[7] solace.com, MQTT Control Packet format. http://docs.solace.com/mqtt-311-prtl-conformance-spec/mqtt2016.

[8] eclipse.org, MQTT and CoAP, IoT Protocols. https://eclipse.org.