مستندات بررسی CoAP

استاد

دکتر فصحتی

نویسنده

کیهان مسعودی

دانشگاه صنعتی شریف

زمستان ۱۴۰۳

فهرست

گذرگاه Constrained Application Protocol

مقدمه

پروتکل کاربرد محدود یا CoAP یک پروتکل وب سبک و بهینه است که برای دستگاه‌ها و شبکه‌های محدود طراحی شده است. این پروتکل به طور خاص برای کاربردهای اینترنت اشیا (IoT) توسعه یافته است و مصرف توان پایین و استفاده بهینه از پهنای باند را تضمین می‌کند.

هدف توسعهCoAP

پروتکل CoAP با اهداف زیر توسعه یافته است:

- **ارتباط مؤثر در شبکه‌های با پهنای باند پایین و تأخیر بالا** :این پروتکل امکان تبادل داده‌ها در شرایطی که پهنای باند محدود و تأخیر بالا است را فراهم می‌کند.
- **مصرف کم منابع** :با طراحی بهینه خود، CoAPنیاز به توان پردازشی و حافظه کمی دارد که برای دستگاه‌های محدود مناسب است.
- **یکپارچگی آسان با فناوری‌های وب** CoAP :با معماری RESTful سازگار است و امکان استفاده مستقیم از فناوری‌های وب را میسر می‌کند.
- **قابلیت ارسال اطلاعات به چندین دستگاه به طور همزمان** :این پروتکل می‌تواند داده‌ها را به چندین دستگاه به صورت همزمان ارسال کند، که این ویژگی برای کاربردهای IoT بسیار مفید است.

ویژگی های CoAP:

ساختار طراحی CoAP :

- **معماری بهینه** CoAP :**RESTful** از متدهای استانداردماننGET ، POST، PUTو DELETE پشتیبانی می‌کند.
- **انتقال بلوکی داده‌ها** :**(Block-wise Transfer)** این قابلیت امکان انتقال داده‌های بزرگ را در قالب بلوک‌های کوچک فراهم می‌کند و باعث افزایش کارایی می‌شود.
- **پشتیبانی از امنیت اختیاری** :از طریق پروتکل (Datagram Transport Layer Security) DTLS امنیت داده‌ها تضمین می‌شود.
- **کشف منابع** :**(Resource Discovery)** این پروتکل امکان کشف خودکار منابع موجود در دستگاه‌ها را فراهم می‌کند.

- **ذخیره‌سازی داده‌های تکراری** :**(Caching)** داده‌های تکراری را ذخیره می‌کند تا نیاز به انتقال مجدد کاهش یابد و کارایی افزایش یابد.

انواع ارتباطات و کدگذاری در CoAP:

- **ارتباط سریال** :پیام‌های CoAP به صورت سریالی منتقل می‌شوند که باعث سادگی انتقال می‌شود.
- **کدگذاری** CoAP :از فرمت دودویی فشرده برای کاهش سربار و افزایش کارایی استفاده می‌کند.
- **تولید سیگنال** :انتقال داده‌ها از طریق بسته‌های UDP انجام می‌شود که ارتباطات غیرهم‌زمان و با تأخیر پایین را ممکن می‌سازد.

اتصال دستگاه‌ها و آدرس‌دهی:

- **قابلیت چند دستگاهی** CoAP :امکان ارتباط چندین دستگاه با یک سرور را فراهم می‌کند.
- **آدرس‌دهی و مسیریابی** :
- **Ip addressing** : این پروتکل از آدرس‌های IP برای شناسایی و ارتباط با دستگاه‌ها استفاده می‌کند.
- **عدم وجود ویژگی‌های مسیریابی داخلی** :مسیریابی توسط پروتکل‌های شبکه زیربنایی مانند RPL مدیریت می‌شود.

مدیریت جریان داده:

- **کنترل جریان** :پیام‌ها در CoAP با استفاده از تأیید پیام (Acknowledgment) مدیریت می‌شوند تا اطمینان حاصل شود که پیام‌ها به مقصد رسیده‌اند.
- **حذف پیام‌های تکراری** :این پروتکل پیام‌های تکراری را شناسایی کرده و نادیده می‌گیرد تا از پردازش بیهوده جلوگیری شود.
- **مکانیزم ارسال مجدد** :در صورت از دست رفتن بسته‌ها، CoAP از مکانیزم ارسال مجدد پیام‌ها استفاده می‌کند تا اطمینان حاصل شود که داده‌ها به درستی به مقصد می‌رسند.

تشخیص خطا و تصحیح خطا:

انواع تشخیص خطا:

۱. **لایه پیوند داده(Data Link Layer)** : یکپارچگی داده‌ها در این لایه از طریق محاسبات چک‌سام UDP حفظ می‌شود.
۲. **لایه‌های بالاتر**: خطاهای در سطح کاربرد از طریق کدهای پاسخ (مانند ۴.۰۴ برای عدم یافتن منبع) شناسایی می‌شوند.

در ارتباط با موضوع تصحیح خطا در پروتکل CoAP بیشتر بر تشخیص خطا و ارسال مجدد پیام تمرکز شده است و استفاده‌ای از مکانیزم‌های پیچیده اصلاح خطا نشده است.

پیام‌ها در CoAP:

در پروتکل **CoAP** ، پیام‌ها برای برقراری ارتباط بین کلاینت و سرور در دستگاه‌های محدود استفاده می‌شوند. CoAP از مدل درخواست-پاسخ (request-response) مشابه HTTP بهره می‌برد اما بهینه‌سازی شده برای شبکه‌های کم‌مصرف و دستگاه‌های IoT است.

اجزای پیام‌های CoAP

پیام‌های CoAP مطابق شکل شامل اجزای زیر هستند:

نسخه(version): این بخش نسخه پروتکل CoAP را مشخص می‌کند که با ۲ بیت مشخص می‌شود.

نوع (Type) : نوع پیام را مشخص می‌کند .

درخواست (Request) :

$$confirmable \rightarrow 00 \quad , \quad Non-confirmable \rightarrow 01$$

پاسخ (Response) :

$$acknowledgement \rightarrow 10 \quad , \quad reset \rightarrow 11$$

طول توکن (Token Length) : این بخش طول توکن را به بایت مشخص می کند که می تواند بین ۰ تا ۸ بایت باشد.

کد (Code) : این بخش کد درخواست یا پاسخ را مشخص می‌کند.

- Method: 0.XX
  - 0. EMPTY
  - 1. GET
  - 2. POST
  - 3. PUT
  - 4. DELETE
  - 5. FETCH
  - 6. PATCH
  - 7. iPATCH
- Success: 2.XX
  - 1. Created
  - 2. Deleted
  - 3. Valid
  - 4. Changed
  - 5. Content
  - 31. Continue

- Client Error: 4.XX
  - 0. Bad Request
  - 1. Unauthorized
  - 2. Bad Option
  - 3. Forbidden
  - 4. Not Found
  - 5. Method Not Allowed
  - 6. Not Acceptable
  - 8. Request Entity Incomplete
  - 9. Conflict
  - 12. Precondition Failed
  - 13. Request Entity Too Large
  - 15. Unsupported Content-Format

- Server error: 5.XX
  - 0. Internal server error
  - 1. Not implemented
  - 2. Bad gateway
  - 3. Service unavailable
  - 4. Gateway timeout
  - 5. Proxying not supported
- Signaling Codes: 7.XX
  - 0. Unassigned
  - 1. CSM
  - 2. Ping
  - 3. Pong
  - 4. Release
  - 5. Abort

شناسه پیام (Message ID) : شناسه پیام یک عدد ۱۶ بیتی است که نقش مهمی در مدیریت ارتباطات CoAP ایفا می‌کند. وظایف اصلی آن شامل موارد زیر است:

- **شناسایی پیام‌های تکراری**:

  اگر یک پیام به دلیل مشکلات شبکه مجدداً دریافت شود، از این شناسه برای تشخیص پیام‌های تکراری استفاده می‌شود.

- **همگام‌سازی پیام‌ها**:

  این شناسه به کلاینت و سرور کمک می‌کند پیام‌های تأییدی (Acknowledgement) یا بازنشانی (Reset) را به پیام‌های اولیه (Confirmable/Non-confirmable) مرتبط کنند.

به دلیل طول ۱۶ بیتی این شناسه، محدوده مقادیر آن از ۰ تا ۶۵۵۳۵ است.

گزینه‌ها (Options) : فرمت گزینه‌ها در CoAP برای انتقال اطلاعات اضافی مانند نوع محتوا، مسیر یا تنظیمات خاص استفاده می‌شود. هر گزینه دارای بخش‌های مختلفی است که به شرح زیر هستند:

Option Delta:

این مقدار فاصله (Delta) بین شناسه گزینه فعلی و گزینه قبلی را مشخص می‌کند:

۰ تا ۱۲ :مقدار دقیق دلتا بین گزینه‌ها، بدون نیاز به مقدار افزونه دلتا.

۱۳:برای دلتاهای بین ۱۳ تا ۲۶۸، مقدار افزونه دلتا یک عدد ۸ بیتی است که برابر دلتا منهای ۱۳ است.

۱۴:برای دلتاهای بین ۲۶۹ تا ۶۵,۸۰۴، مقدار افزونه دلتا یک عدد ۱۶ بیتی است که برابر دلتا منهای ۲۶۹ است.

۱۵:رزرو شده برای نشانگر payload که در آن دلتا و طول گزینه برابر 0xFF می‌شوند.

Option Length:

مشخص می‌کند که طول مقدار گزینه چقدر است:

۰ تا ۱۲ :طول دقیق گزینه، بدون نیاز به مقدار افزونه طول.

۱۳:برای طول بین ۱۳ تا ۲۶۸، مقدار افزونه طول یک عدد ۸ بیتی است که برابر طول منهای ۱۳ است.

۱۴:برای طول بین ۲۶۹ تا ۶۵,۸۰۴، مقدار افزونه طول یک عدد ۱۶ بیتی است که برابر طول منهای ۲۶۹ است.

۱۵:رزرو شده برای استفاده‌های آینده؛ تنظیم این مقدار به 0xFFخطا محسوب می‌شود.

Option Value:

اندازه این بخش بر اساس مقدار Option Length تعیین می‌شود و حاوی داده اصلی گزینه است.

معنای دقیق و فرمت این بخش به گزینه خاص مربوط بستگی دارد.

بار پیام (Payload) : این قسمت شامل داده‌های اصلی پیام است.

Octet offset		0								1								2								3							
	Bit offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
4	32	ver		type		token length				request/response code								message ID															
8	64	token (0–8 bytes)																															
12	96																																
16	128	options (if available)																															
20	160	1	1	1	1	1	1	1	1	payload (if available)																							

شکل ۱: فرمت پیام CoAP

انواع پیام‌هایCoAP:

۱. **پیام تأییدپذیر(Confirmable)**

این پیام‌ها نیازمند دریافت تأییدیه (Acknowledgment) هستند.اگر کلاینت یک پیام تأییدپذیر ارسال کند اما پاسخی از سرور دریافت نکند، پیام مجدداً ارسال می‌شود. استفاده از تأییدیه باعث افزایش اطمینان از تحویل موفق پیام می‌شود.
این نوع پیام‌ها برای انتقال اطلاعات حیاتی و مهم استفاده می‌شوند، مانند فرمان‌های کنترلی یا داده‌های حساس حسگرها.
مثال :ارسال دستور روشن کردن یا خاموش کردن یک دستگاهIoT .

۲. **پیام غیرتأییدپذیر(Non-confirmable)**

پیام‌هایی که نیازی به دریافت تأییدیه ندارند. این پیام‌ها معمولاً برای داده‌های کم‌اهمیت یا زمانی که تأخیر در شبکه اهمیتی ندارد استفاده می‌شوند. در صورت عدم تحویل یا از دست رفتن پیام، CoAP اقدام به ارسال مجدد نمی‌کند.
این نوع پیام‌ها به دلیل سبک بودن برای ارتباطات سریع و بدون بار اضافی مناسب هستند.
مثال :ارسال داده‌های دما یا رطوبت که به‌روزرسانی مداوم دارند.

۳. **پیام تأییدیه(Acknowledgment)**

این پیام‌ها در پاسخ به پیام‌های تأییدپذیر (Confirmable) ارسال می‌شوند. هنگامی که سرور یک پیام تأییدپذیر دریافت می‌کند، یک پیام تأییدیه برای اعلام موفقیت دریافت ارسال می‌کند. پیام تأییدیه ممکن است شامل داده مرتبط با درخواست نیز باشد. اگر کلاینت این پیام را دریافت نکند، اقدام به ارسال مجدد پیام تأییدپذیر می‌کند.
مثال :تأیید موفقیت‌آمیز دریافت فرمان روشن کردن یک دستگاه.

۴. **بازنشانی(Reset)**

پیام‌هایی که نشان می‌دهند یک پیام دریافت شده، اما امکان پردازش آن وجود ندارد. این پیام معمولاً در مواردی ارسال می‌شود که پیام دریافتی نامعتبر باشد یا سرور توانایی پردازش آن را نداشته باشد. بازنشانی می‌تواند به دلایلی مانند شناسه پیام اشتباه، مشکلات در فرمت پیام یا ناسازگاری گزینه‌ها رخ دهد.

مثال :سرور یک پیام ناشناخته یا با شناسه نادرست دریافت کند و پیام بازنشانی ارسال کند تا کلاینت را از این خطا مطلع سازد.

نتیجه‌گیری:

پروتکل CoAP با طراحی سبک و بهینه خود به یکی از راه‌حل‌های مناسب برای ارتباطات در دستگاه‌های محدود و شبکه‌های IoT تبدیل شده است. ویژگی‌هایی مانند کدگذاری فشرده، پشتیبانی از انتقال بلوکی داده، مکانیسم‌های تشخیص و اصلاح خطا و امنیت اختیاری، این پروتکل را به گزینه‌ای کارآمد برای کاربردهای مختلف از جمله مدیریت انرژی هوشمند، اتوماسیون خانگی و نظارت محیطی تبدیل کرده است.