عنوان: مدیریت و اینترنت اشیاء

چکیده

مطالعه و توسعه‌ی برنامه‌های کاربردی اینترنت اشیاء (IoT)، وب و تلفن همراه، رو به افزایش است. نرم‌افزارهای کاربردی، با داده‌های سروکار دارند که از حوزه‌های مختلفی از قبیل حمل و نقل، خانه‌های هوشمند، مراقبت از سلامتی، سرویس‌های عمومی، صنعت و بسیاری دیگر بدست می‌آید. مطالعات قبلی بر روی مدیریت داده‌های بدست آمده تمرکز داشتند. با این حال، مدیریت منابع ناهمگنی که داده‌ها را بدست می‌آورند، حوزه‌ای است که نیازمند توجه بیشتری می‌باشد. این مقاله مدیریت منابع در اینترنت اشیاء را بررسی می‌کند. این هدف به وسیله‌ی ارائه‌ی یک لایه‌ی لبه‌ای مجازی منابع تحقق یافته است، که دسترسی و پیکربندی به منابع فیزیکی محدود را قادر می‌سازد. معماری ارائه شده بر روی استفاده از منابع مجازی به عنوان یک مفهوم مدیریتی تمرکز می‌کند و رویکردهای مختلف را در ارزیابی کارایی وسایل محاسباتی لبه‌ای شناسایی می‌کند. با استفاده از پروتکل اینترنت اشیاء CoAP، منابع مجازی در لبه‌ی شبکه قرار گرفته‌اند. یک ارزیابی از یک منبع مجازی Go CoAP ارائه شده است.

کلمات کلیدی: اینترنت اشیاء؛ مدیریت IoT؛ منابع مجازی؛ CoAP

۱- مقدمه

در اینترنت اشیاء[۱و۲]، ارتباطات و مدیریت منابع بر اساس یک تکنولوژی انجام شده است که در آن شبکه محدود پیاده‌سازی شده است[۶و۷و۸]. هر فروشنده از توسعه‌ی محرمانه-سازگاری و اختصاصی استانداردهای ارتباطی خود مطمئن شده است. برنامه‌های کاربردی به وسیله‌ی استانداردهای اختصاصی پیشتیبانی شده‌اند تا اطلاعات قابل اعتمادی را به شیوه‌ای کارآمد و امن از محیط‌های محدود در ابر بدست آورند[۳،۴و۵]. این امر کاربران را تحت تاثیر قرار می‌دهد تا تکنولوژی خاصی را در کل پیاده‌سازی‌ها جهت ساده نمودن وظایف مدیریتی قبول کنند (ITU-T M.3400). تعدادی از مطالعات با تمرکز بر روی وسایل فیزیکی اینترنت اشیاء وجود دارند، ولی هدف اصلی آنها ساده نمودن کار توسعه‌دهندگان است[۱۰و۱۱]. امروزه، راه‌حلی برای اجازه‌ی دسترسی جهت متنوع ساختن منابع شبکه‌ای اینترنت اشیاء برای بدست آوردن اطلاعات آنها وجود ندارد. این امر باعث محدودیت‌هایی در انجام دیدگاه اینترنت اشیاء می‌شود[۹]. این مقاله علاوه بر دیدگاهی که اشیاء ناهمگن در آن متصل هستند، بلکه بدست آوردن اطلاعات را به صورت بی‌درنگ از اشیائی که در حال کار در محیط هستند نیز گسترش می‌دهد. معماری ارائه شده در این مقاله بر روی خلاصه نمودن پیچیدگی دسترسی و پیکربندی منابع شبکه‌های

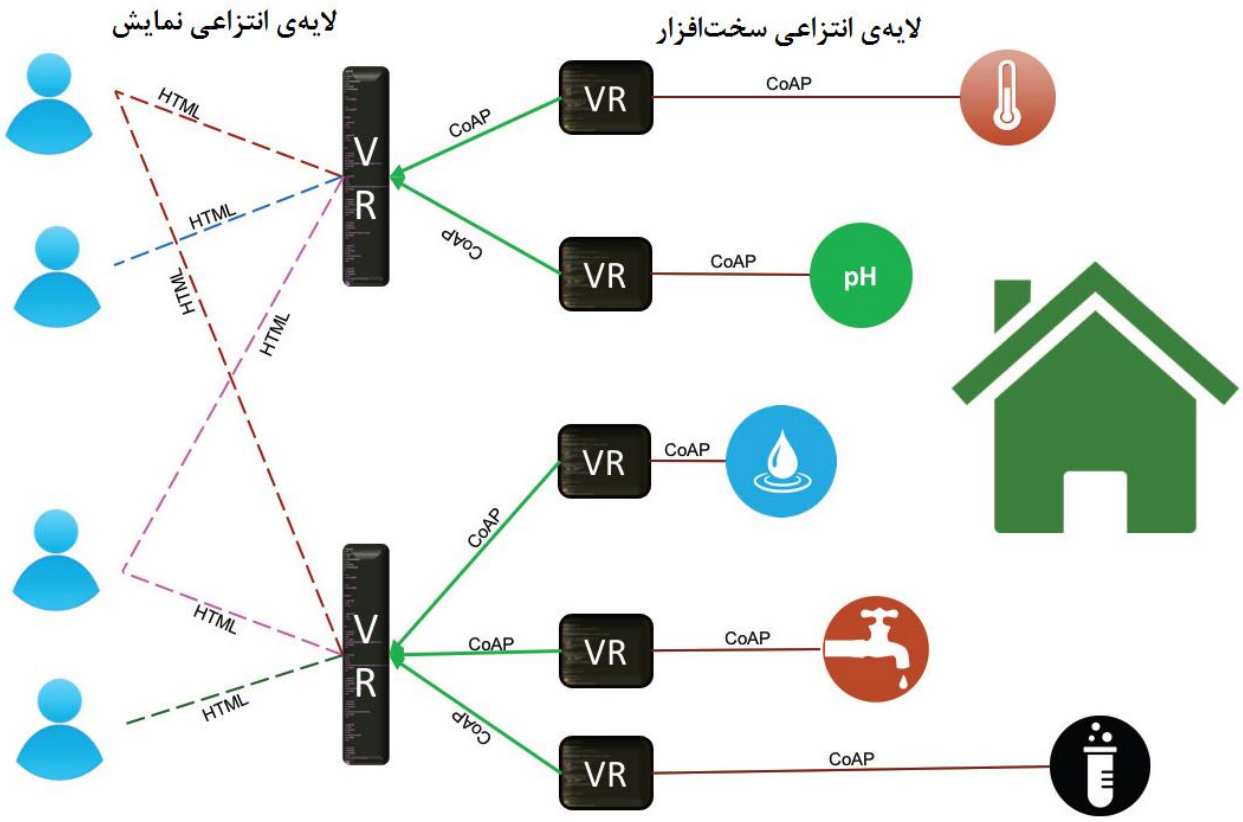
محدود ناهمگن به مفهوم منابع مجازی متمرکز شده است و یک گزینه‌ی مدیریتی ساده شده‌ای را پیشنهاد می‌دهد. این مقاله ایجاد منابع مجازی در حال اجرا بر روی وسایل محاسباتی لبه‌ای را ارائه می‌دهد. همچنین، این مطالعه از پروتکل برنامه‌ی کاربردی محدود[1] (CoAP) جهت ارتباط منابع مجازی و از زبان برنامه‌نویسی Go برای ایجاد آنها استفاده کرده است. این نمایش اجازه می‌دهد تا اجزای محدود به راحتی توسط اشخاص ثالث قابل دسترسی باشند. ارتباط مستقیم با منابع فیزیکی به وسیله‌ی معرفی لایه‌های منابع مجازی شکسته می‌شود.

ادامه‌ی این مقاله به شرح زیر سازماندهی شده است. بخش ۲ تعریفی از منابع اینترنت اشیاء ما را ارائه می‌دهد. بخش ۳ راه‌اندازی آزمایشی را توضیح می‌دهد. بخش ۴ ارزیابی کارایی نمونه‌ی اولیه‌ی ما را ارائه داده و نتیجه‌گیری مقاله به همراه یک خلاصه در بخش ۵ می‌باشد.

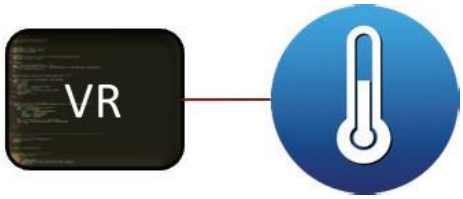
۲- طراحی معماری منابع مجازی اینترنت اشیاء

ما یک منبع مجازی اینترنت اشیاء را به عنوان یک مفهوم انتزاعی از دیگر منابع، چه مجازی یا غیر، تعریف می‌کنیم. طراحی معماری منابع مجازی اینترنت اشیاء به صورت گرافیکی در شکل ۱ نمایش داده شده است. با استفاده از ویژگی‌های REST در CoAP (RFC 7252)، یک منبع مجازی وضعیت و روش‌های خود را نمایش می‌دهد که با حالت‌ها در تعامل بوده و می‌تواند تعاملات بین منابع ترکیبی را حل کند. کل معماری شامل سه لایه‌ی اصلی است که عبارتند از: لایه‌ی انتزاعی نمایش[2] (VAL)، لایه‌ی انتزاعی سخت‌افزار[3] (HAL) و لایه‌ی فیزیکی. لایه‌ی فیزیکی حسگرهایی را نمایش می‌دهد که در حال کار در محیط هستند. دو لایه، انتزاعی سخت‌افزار و انتزاعی نمایش، در واقع میان‌افزارهای برنامه‌های کاربردی قرار گرفته در لبه هستند، که با هدف کاهش پیچیدگی دسترسی و پیکربندی منابع فیزیکی می‌باشند. HAL پایین‌ترین سطح پیوستگی را بین یک منبع فیزیکی و یک منبع مجازی نشان می‌دهد که به یک رابطه‌ی یک به یک منجر می‌شود، شکل ۲ را ببینید. منابع مجازی که در این لایه قرار گرفته‌اند از افعال CoAP برای دریافت سفارشات جهت انجام عملیات بر روی منابع فیزیکی که با آن مواجه هستند، استفاده می‌کنند.

[1] Constrained Application Protocol (CoAP)
[2] View Abstraction Layer (VAL)
[3] Hardware Abstraction Layer (HAL)



شکل ۱: طراحی معماری منابع مجازی اینترنت اشیاء. مثالی از اکوسیستم اینترنت اشیاء در یک گلخانه به همراه منابع مجازی.

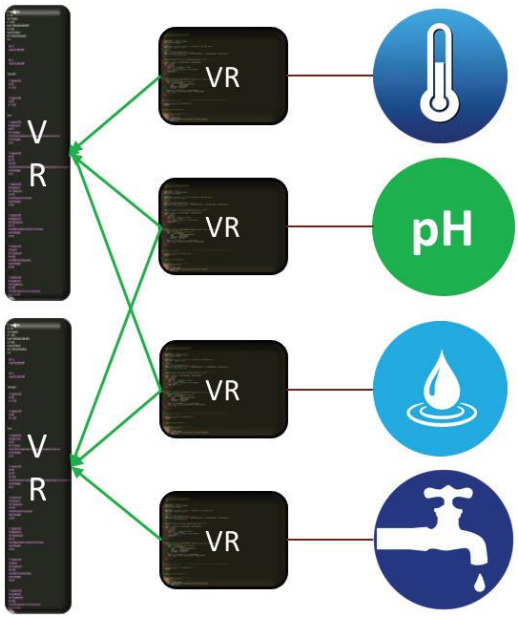


شکل ۲: رابطه‌ی یک-به-یک منبع مجازی.

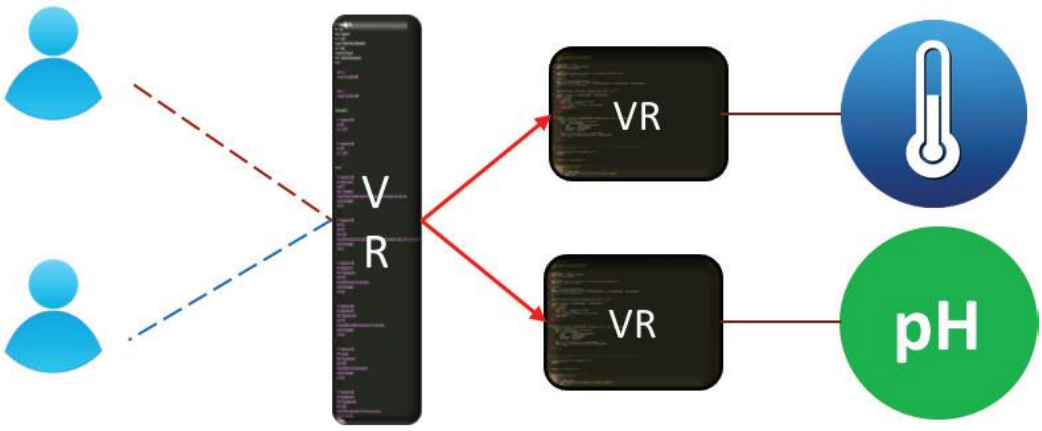
VAL به عنوان یک پل بین کاربران نهایی و حسگرها عمل می‌کند. VAL یک پیوستگی بین یک یا چند منبع مجازی HAL و یک یا چند منبع مجازی VAL را نشان می‌دهد، که یک رابطه‌ی چند-به-چند را نتیجه می‌دهد، شکل ۳ را ببینید. هدف منابع مجازی که در این لایه قرار دارند در واقع فراهم نمودن نمایش‌های پویا برای انواع مختلفی از کاربران با اولویت‌های دسترسی متفاوت است. همچنین، این لایه به عنوان یک مرکز پردازشی-لبه‌ای عمل می‌کند که در آن ردیف-داده‌ها بدون رفتن به ابر ارزیابی می‌شوند. در این لایه، منابع مجازی به دو گروه، منابع مجازی بدون-حالت و با حالت-کامل تقسیم می‌شوند. منابع مجازی بدون-حالت،

شکل ۴ را مشاهده کنید، از الگوی CoAP REST برای بیرون کشیدن حالت‌ها در هر زمان که درخواستی دریافت شد، استفاده می‌کنند. منابع مجازی با حالت-کامل، به شکل ۵ نگاه کنید، حالت‌های منابع ترکیبی خود را در یک پایگاه‌داده نگهداری می‌کنند و از الگوی مشاهده‌ی CoAP برای دریافت بروزرسانی‌های منابعی که در حال مشاهده هستند، استفاده می‌کنند.

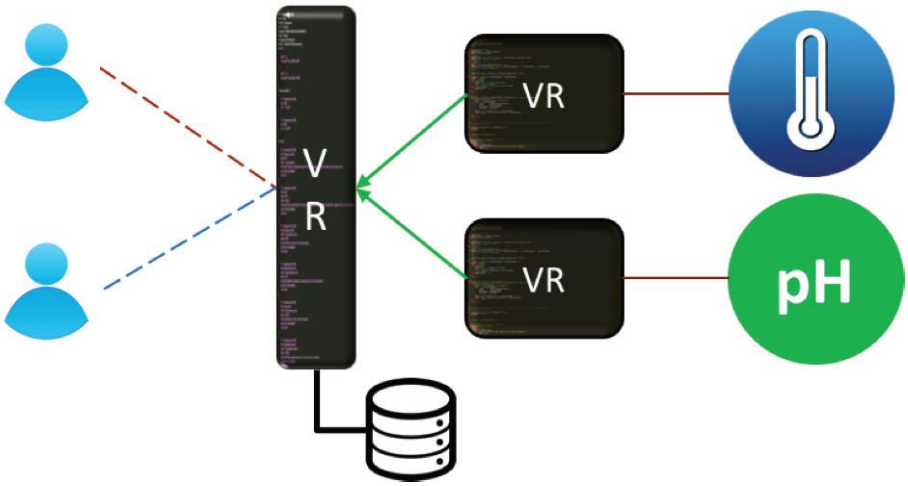
این معماری برای عبور از مانع داشتن تنها یک سطح ارتباطی، فیزیکی، در هنگام کار با حسگرها در نظر گرفته شده است.



شکل ۳: رابطه‌ی چند-به-چند منابع مجازی.



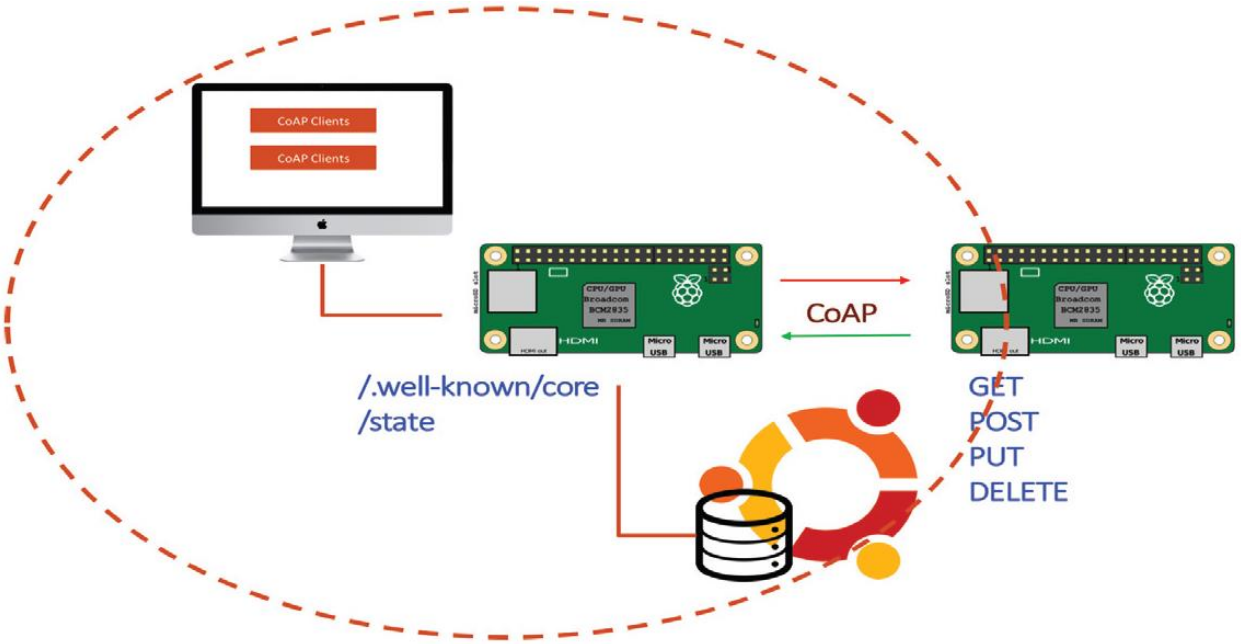
شکل ۴: نمایش یک منبع مجازی بدون حالت.



شکل ۵: نمایش یک منبع مجازی با حالت کامل.

۳- راه‌اندازی آزمایشی

این آزمایش بر روی یک منبع VAL اجرا شد. نمونه‌ی اولیه شامل یک زیرساخت شبکه‌ی محدود، منابع مجازی و درخواست‌کنندگان شبیه‌سازی شده می‌باشد. هنگامی که منبع VAL یک درخواست CoAP را از درخواست‌کننده‌ی CoAP دریافت می‌کند، منبع VAL حالت منابع ترکیبی خود را می‌گیرد و اطلاعات پردازش شده را با یک پیام CoAP ارسال می‌کند. یک Raspberry Pi با زبان GO و پروتکل CoAP به عنوان لایه‌ی محاسبات لبه‌ای برای اجرای منابع مجازی استفاده شده است، شکل ۶ را مشاهده کنید. ما منبع مجازی و درخواست‌کنندگان را با استفاده از ویژگی‌های روال‌ها و کانال‌های GO ایجاد نموده‌ایم، شکل ۷ را ببینید. ما معاری پیشنهادی را به وسیله‌ی وسایلی با مشخصات زیر راه‌اندازی کردیم: Raspberry PI 2 مدل B. پردازنده ARMv7، CPU: ARM Cortex-A7 چهار هسته‌ای- ۹۰۰ مگاهرتز، حافظه‌ی RAM: ۱ گیگابایت، سیستم عامل: Raspbian OS. IMac با پردازنده: Intel Core i7- ۳,۵ گیگاهرتز. حافظه‌ی RAM: ۱۶ گیگابایت، سیستم عامل ۶۴ بیتی. درخواست‌کنندگان با استفاده از یک اتصال WiFi اختصاصی به منبع مجازی متصل شدند.



شکل ۶: راه‌اندازی آزمایشی

```go
func handle_request(m *coap.Message) (payload []byte, err error) {
    code := m.Code
    switch code {
    case coap.GET:
        payload, err = handle_GET("temperature")
    case coap.DELETE:
        chan_stop <- true
    }
    return
}
```

```go
func listening_to_exit() {
    for {
        select {
        case <-chan_stop:
            log.Println("I am leaving...  god bye")
            os.Exit(0)
        }
    }

}
```

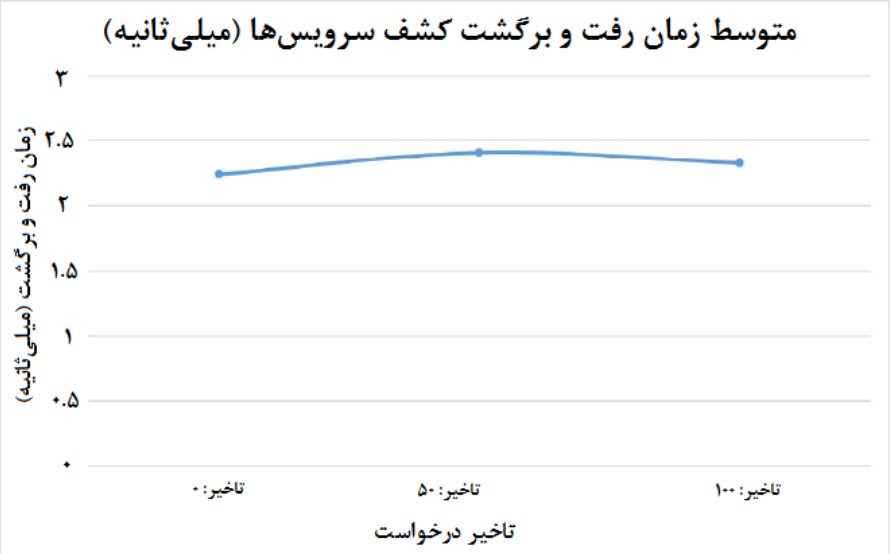
شکل ۷: نمونه کدی از زبان GO

۴- ارزیابی

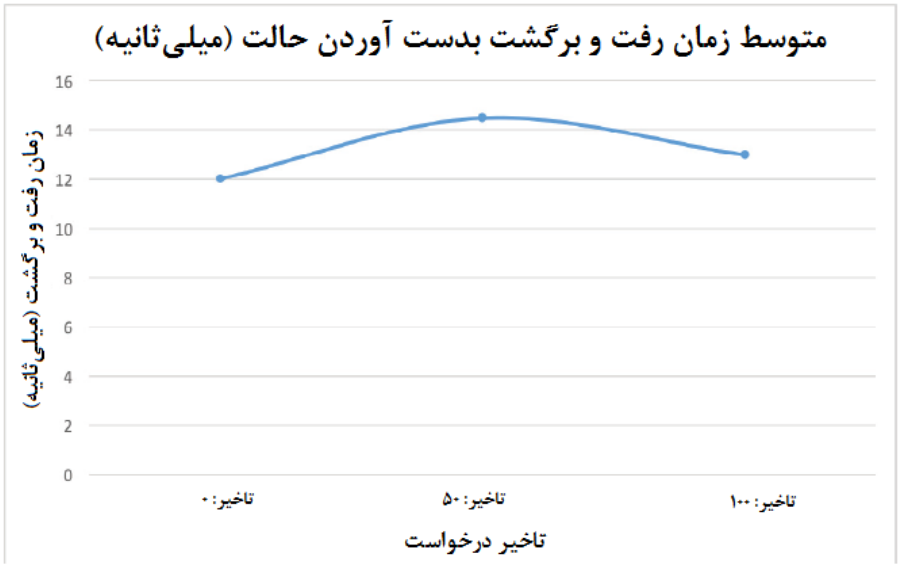
برای ارزیابی مفید بودن منبع مجازی خود، توان عملیاتی و زمان‌های پاسخ آنها را اندازه‌گیری نمودیم. ما رفتار منبع مجازی را با آزمایش دو فرآیند، فرمت لینک هسته[۴] (RFC 6690) کشف سرویس‌ها از طریق یک واسط شناخته شده و بازیابی حالت جاری ارزیابی نمودیم. این ارزیابی‌ها از دیدگاه درخواست‌کننده انجام شده است. همچنین، ما زمان اتصال پایگاه‌داده را در منبع با حالت کامل VAL نیز ارزیابی نموده‌ایم. سرویس‌هایی که منبع مجازی ما عرضه می‌کند به عنوان سرویس‌های CoAP REST میزبانی شده‌اند. سناریو شامل یک مبنع مجازی GO CoAP در حال اجرا بر روی یک Raspberry Pi و هزار درخواست‌کننده‌ی مجازی Go CoAP در حال اجرا بر روی کامپیوتر IMac می‌باشد. این آزمایش زمان‌های تاخیر در درخواست‌های درخواست‌کنندگان را از ۰ تا ۱۰۰ میلی‌ثانیه، در بازه‌های ۵۰ میلی‌ثانیه نشان می‌دهد که سطوح مختلف همزمانی را شبیه‌سازی می‌کند.

نتایج برای تمام هزار درخواست در هر سناریوی تاخیر در ادامه شرح داده شده است. متوسط زمان رفت و برگشت[۵] (RTT) کشف سرویس برای سه مورد تاخیر یعنی ۰، ۵۰ و ۱۰۰ میلی‌ثانیه به ترتیب برابر با ۲,۲۴ میلی ثانیه، ۲,۴۰ میلی‌ثانیه و ۲,۳۳ میلی‌ثانیه است، شکل ۸ را مشاهده کنید. در این آزمایش، بارکاری منبع مجازی و زمان‌های پاسخ آن برای سه مورد تاخیر ثابت باقی می‌ماند. متوسط زمان رفت و برگشت (RTT) برای بدست آوردن حالت جاری منبع مجازی برای سه مورد تاخیر برابر با ۱۲,۰۱ میلی‌ثانیه، ۱۴,۴۷ میلی‌ثانیه و ۱۲,۹۸ میلی‌ثانیه است. در این مرحله، متوسط زمان پاسخ افزایش یافت چرا که این یک منبع مجازی با حالت کامل بوده و مجبور است به پایگاه‌داده برای بدست آوردن حالت جاری منابع ترکیبی خود مراجعه کند، شکل ۹ را ببینید. متوسط زمان پاسخی که منبع مجازی برای رفتن به پایگاه‌داده جهت جستجوی حالت یک منبع ترکیبی خاص صرف می‌کند در سه زمان تاخیر برابر با ۷,۵۳ میلی‌ثانیه، ۸,۲۹ میلی‌ثانیه و ۸,۰۷ میلی‌ثانیه است، شکل ۱۰ را ببینید. در هر سه ارزیابی، نقاط اوجی وجود داشت که می‌توانند به ترافیک شبکه‌ای و فرآیندهای پشتیبانی نسبت داده شود و ناهنجاری‌های رایجی را نشان می‌دهند. پایگاه‌داده در یک ماشین جدا اجرا شد.

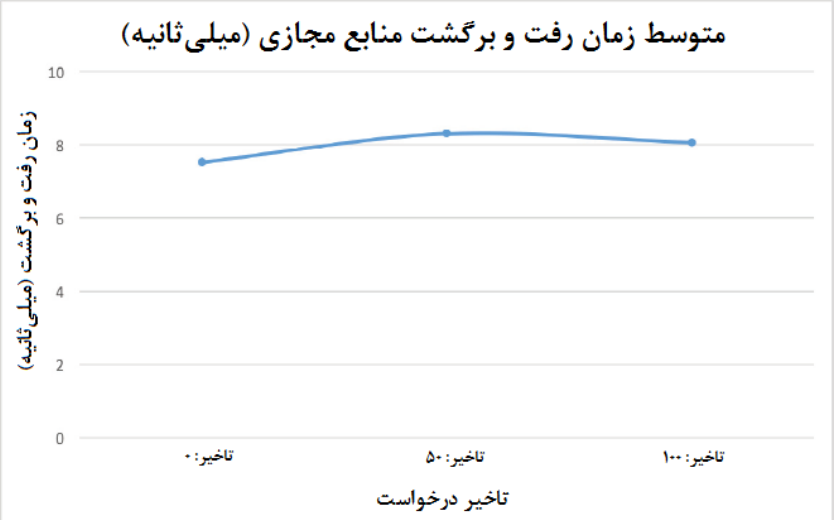
[4] Core Link Format
[5] Round Trip Time (RTT)



شکل ۸: هزار درخواست‌کننده‌ی مجازی. زمان رفت و برگشت فرآیند کشف سرویس منبع مجازی.



شکل ۹: هزار درخواست‌کننده‌ی مجازی. زمان رفت و برگشت فرآیند بدست آوردن حالت منبع مجازی.



شکل ۱۰: هزار درخواست‌کننده‌ی مجازی. زمان رفت و برگشت اتصال به پایگاه‌داده‌ی منبع مجازی.

۵- نتیجه‌گیری

این مقاله پیچیدگی دسترسی و پیکربندی منابع اینترنت اشیاء را در یک لایه‌ی محاسباتی لبه‌ای خلاصه می‌کند که از طریق پروتکل CoAP مدیریت می‌شود. منبع مجازی ما به صورتی که انتظار می‌رفت اجرا شد و بـه تمـام درخواست‌ها بدون توجه به سطح همزمانی که با آنها روبرو بود، پاسخ داد. لایه‌ی منابع مجازی توانست بـین هـر جزء ترکیبی به راحتی تعامل برقرار کند همانطور که به راحتی برای مصرف یک سـرویس REST کامـل انجـام داد، در این مورد یک سرویس CoAP REST کامل موردنظر است. معماری ارائه شده از یـک راه‌حـل اقتصـادی تشکیل می‌شود چنانکه ارزیابی داده‌ها در لبه‌ی شبکه‌ی اینترنت اشیاء انجام می‌شود و تنها اطلاعات مفید به ابر منتقل می‌شوند. این اولین گام در ساخت یک محیط محاسباتی لبه‌ای است که در آن منابع مجـازی بـه عنـوان یک گزینه‌ی توصیه شده برای مدیریت شبکه‌های متنوع اینترنت اشیاء معرفی می‌شوند. بنابراین، مـا مـی‌تـوانیم چشم‌انداز اینترنت اشیاء را به انجام برسانیم.

به عنوان یک کار آینده، این پژوهش تعداد زیادی حسگر و محرک آزمایش شده را برای انـدازه‌گیـری کاربردهای کلی معرفی خواهد نمود. همچنین، مقاله شامل منابع مجازی توزیع شده نیز خواهد بود.

مراجع

[1] Ashton, K. That internet of things thing. *RFiD Journal 22*, 7 (2009), 97–114.

[2] Atzori, L., Iera, A., and Morabito, G. The Internet of Things: A survey. *Computer Networks 54*, 15 (2010), 2787–2805.

[3] Bandyopadhyay, S., and Bhattacharyya, A. Lightweight Internet protocols for web enablement of sensors using constrained gateway devices. *2013 International Conference on Computing, Networking and Communications, ICNC 2013* (2013), 334–340.

[4] Al-Fuqaha, A., Guizani, M.,Mohammadi, M., Aledhari, M., and Ayyash,M. Internet of Things: A Survey on Enabling Technologies, Protocols and Applications. *IEEE Communications Surveys & Tutorials PP*, 99 (2015), 1–1.

[5] Evdokimov, S., Fabian, B., Kunz, S., and Schoenemann, N. Comparison of discovery service architectures for the internet of things.

[6] Castellani, A. P., Gheda, M., Bui, N., Rossi, M., and Zorzi, M. Web services for the internet of things through coap and exi. In *2011 IEEE International Conference on Communications Workshops (ICC)* (June 2011), pp. 1–6.

[7] Nati, M., Gluhak, A., Abangar, H., and Headley, W. SmartCampus: A user-centric testbed for Internet of Things experimentSmartCampus: A user-centric testbed for Internet of Things experimentationation. *2013 16th International Symposium on Wireless Personal Multimedia Communications, WPMC 2013 - Co-located with Global Wireless Summit 2013* (2013).

[8] Hakiri, A., Berthou, P., Gokhale, A., and Abdellatif, S. Publish/subscribe-enabled software defined networking for efficient and scalable iot communications. *Communications Magazine, IEEE 53*, 9 (2015), 48–54.

[9] Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems 29*, 7 (sep 2013), 1645–1660.

[10] Kovatsch, M., Mayer, S., and Ostermaier, B. Moving application logic from the firmware to the cloud: Towards the thin server architecture for the internet of things. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on* (2012), IEEE, pp. 751–756.

[11] Mainetti, L., Mighali, V., and Patrono, L. A software architecture enabling the web of things. *Internet of Things Journal, IEEE 2*, 6 (2015), 445–454.