###### به نام خدا



###### پژوهشکده فناوری اطلاعات

مقدمه‌ای بر بستر‌های اینترنت اشیا

پروژه: اینترنت اشیا (طرح فداک)

کد پروژه:

|  |  |
| --- | --- |
| مجری: |  |
| تهیه‌کننده: |  |
| کد گزارش: |  |
| تاریخ ارائه: |  |
| نسخه/وضعیت: |  |

شناسنامه گزارش

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **عنوان:** | | | | **شماره نسخه:** |
| **كد:** | **نوع گزارش:** | | | **تاریخ ارائه گزارش:** |
| **نام پروژه:** | | | **نوع پروژه:** | |
| **تاریخ شروع:** | | | **تاریخ پایان:** | |
| **نام گروه:** | | | | |
| **كد پروژه:** | | | **شماره و تاریخ قرارداد:** | |
| **مجری:** | | | **ناظر/ ناظرین:** | |
| **تهیه كننده/ تهیه كنندگان:** | | | | |
| **نام و نشانی مجری:** | | | | |
| **نام و نشانی حمایت كننده:**  تهران، انتهای خیابان كارگر شمالی، پژوهشگاه ارتباطات و فناوری اطلاعات (مركز تحقیقات مخابرات ایران)\_ كد پستی: 1439955471 \_ تلفن: 10-8005508 | | | | |
| **ملاحظات:** | | | | |
| **چكیده:** | | | | |
| **كلمات كلیدی:** | | | | |
| **وضعیت گزارش:** | | **زبان گزارش:** | | |
| **وضعیت دسترسی:** | | **تعداد صفحات:** | | |

چکیده

افزایش تعداد دستگاه‌هایی که از طریق اینترنت به یکدیگر متصل می‌شوند، نشانه‌هایی از تحقق اینترنت اشیا است. دنیایی که در آن اشیا با حداقل دخالت انسانی با یکدیگر ارتباط دارند و سرویس‌های متنوعی را ارائه می‌کنند. به منظور دستیابی به اهداف اینترنت اشیا جنبه‌های متعددی باید در نظر گرفته شود. در این تحقیق ابتدا به معرفی اینترنت اشیا خواهیم پرداخت. سپس شهرهای هوشمند را معرفی میکنیم و چالش‌های پیش رو برای ایجاد شهرهای هوشمند را بررسی می‌کنیم. در نهایت به بررسی بسترهای موجود برای اینترنت اشیا می‌پردازیم.

کلمات کلیدی: اینترنت اشیا، شهر‌های هوشمند، بسترهای اینترنت اشیا

اطلاعات مرتبط

مستندات مرتبط

|  |  |  |
| --- | --- | --- |
| **شماره مستند** | **نوع مستند** | **نام مستند** |
|  |  |  |
|  |  |  |

تغییرات اعمال‌شده در نسخه‌های پیشین

|  |  |  |
| --- | --- | --- |
| **شماره نسخه** | **تاریخ** | **تغییرات اعمال شده** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

تأییدکنندگان

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **نام و نام خانوادگی** | **تاریخ** | **امضاء** | **ملاحظات** |
| **مجری پروژه** |  |  |  |  |
| **تهیه كننده/  تهیه كنندگان** |  |  |  |  |
| **ناظر پروژه** |  |  |  |  |
| **مدیر گروه** |  |  |  |  |
| **مسئول مستندات پژوهشكده** |  |  |  |  |
| **رئیس پژوهشكده/  معاون پژوهشی** |  |  |  |  |

تقدیر و تشکر

سرفصل مطالب

[فهرست جداول 8](#_Toc465688404)

[1 فصل اول: مرکز تحقیقات مخابرات ایران 10](#_Toc465688405)

[1-1 ماموریت‌های سازمان 10](#_Toc465688406)

[1-2 چارت سازمانی 12](#_Toc465688407)

[2 فصل دوم: مقدمه 13](#_Toc465688408)

[2-1 پیشینه تحقیقاتی 14](#_Toc465688409)

[3 فصل سوم: شهرهای هوشمند 18](#_Toc465688410)

[3-1 مقدمه 18](#_Toc465688411)

[3-2 ویژگی‌ها 19](#_Toc465688412)

[3-3 چالش‌ها 19](#_Toc465688413)

[3-4 نمونه‌های بارز 26](#_Toc465688414)

[4 فصل سوم: بستر‌های اینترنت اشیا 29](#_Toc465688415)

[4-1 مقدمه 29](#_Toc465688416)

[4-2 ۸ قسمت اصلی بستر اینترنت اشیا 30](#_Toc465688417)

[4-2-1 اتصال و نرمال‌سازی 31](#_Toc465688418)

[4-2-2 مدیریت دستگاه‌ها 31](#_Toc465688419)

[4-2-3 پایگاه داده 31](#_Toc465688420)

[4-2-4 پردازش و مدیریت عمل‌ها 32](#_Toc465688421)

[4-2-5 تجزیه و تحلیل 33](#_Toc465688422)

[4-2-6 مجسم‌سازی داده‌ها 33](#_Toc465688423)

[4-2-7 ابزارات اضافه 33](#_Toc465688424)

[4-3 پروتکل‌های ارتباطی 33](#_Toc465688425)

[4-3-1 پروتکل‌های لایه کاربرد 35](#_Toc465688426)

[4-3-2 پروتکل CoAP 35](#_Toc465688427)

[4-3-3 پروتکل MQTT 37](#_Toc465688428)

[4-3-4 پروتکل AMQP 39](#_Toc465688429)

[4-3-5 پروتکل DDS 41](#_Toc465688430)

[4-3-6 پروتکل XMPP 42](#_Toc465688431)

[4-3-7 جمع‌بندی 43](#_Toc465688432)

[5 فصل چهارم 44](#_Toc465688433)

[5-1 بیگ دیتا چیست؟ 44](#_Toc465688434)

[5-2 بیگ دیتا و اینترنت اشیا 45](#_Toc465688435)

[5-3 نحوه برخورد مناسب با بیگ دیتا 46](#_Toc465688436)

[5-4 هدوپ 46](#_Toc465688437)

[5-4-1 اهمیت هدوپ 47](#_Toc465688438)

[5-4-2 چالش های استفاده از هادوپ 48](#_Toc465688439)

[5-4-3 اینترنت اشیا و هدوپ 48](#_Toc465688440)

[5-5 اسپارک 49](#_Toc465688441)

[5-5-1 مزیت‌های اسپارک 50](#_Toc465688442)

[5-5-2 اکوسیستم آپاچی اسپارک 51](#_Toc465688443)

[5-6 درباره فلوم 52](#_Toc465688444)

[5-7 پروژه ما 52](#_Toc465688445)

[5-8 کارهای آینده 53](#_Toc465688446)

[5-9 پروژه های انجام شده 53](#_Toc465688447)

# فهرست جداول

[جدول 1 پروتکل‌های لایه کاربرد 43](#_Toc465626731)

فهرست اشکال

[شکل 1 چارت سازمانی پژوشگاه ارتباطات و فناوری اطلاعات 12](#_Toc465641643)

[شکل 2 تقسیم‌بندی حوزه‌های تحقیقاتی مرتبط با اینترنت اشیا 17](#_Toc465641644)

[شکل 3 شمای یک شهر هوشمند 18](#_Toc465641645)

[شکل 4 شهرهوشمند آمستردام 27](#_Toc465641646)

[شکل 5 شبکه‌ی جدید اتوبوس در بارسلونا 27](#_Toc465641647)

[شکل 6 ۸ بلاک سازنده‌ی اصلی یک بستر اینترنت اشیا 30](#_Toc465641648)

[شکل 7 راه‌کار IFTTT برای روشن کردن خودکار لامپ‌ها در صبح 32](#_Toc465641649)

[شکل 8 پروتکل‌های ارتباطی مطرح در اینترنت اشیا 34](#_Toc465641650)

[شکل 9 ساختار پروتکلCoAP 36](#_Toc465641651)

[شکل 10 فرمت کلی پیغام CoAP 37](#_Toc465641652)

[شکل 11 سناریوی پیاده‌سازی CoAP 37](#_Toc465641653)

[شکل 12 معماری MQTT 38](#_Toc465641654)

[شکل 13 ساختار کلی پیغام MQTT 38](#_Toc465641655)

[شکل 14 سناریوی پیاده‌سازی MQTT 39](#_Toc465641656)

[شکل 15 ساختار پیغام در AMQP 40](#_Toc465641657)

[شکل 16 ارسال پیغام به صورتPublish-Subscribe 41](#_Toc465641658)

[شکل 17- معماری مفهومیDDS 42](#_Toc465641659)

[شکل 18- سناریوی XMPP[18] 43](#_Toc465641660)

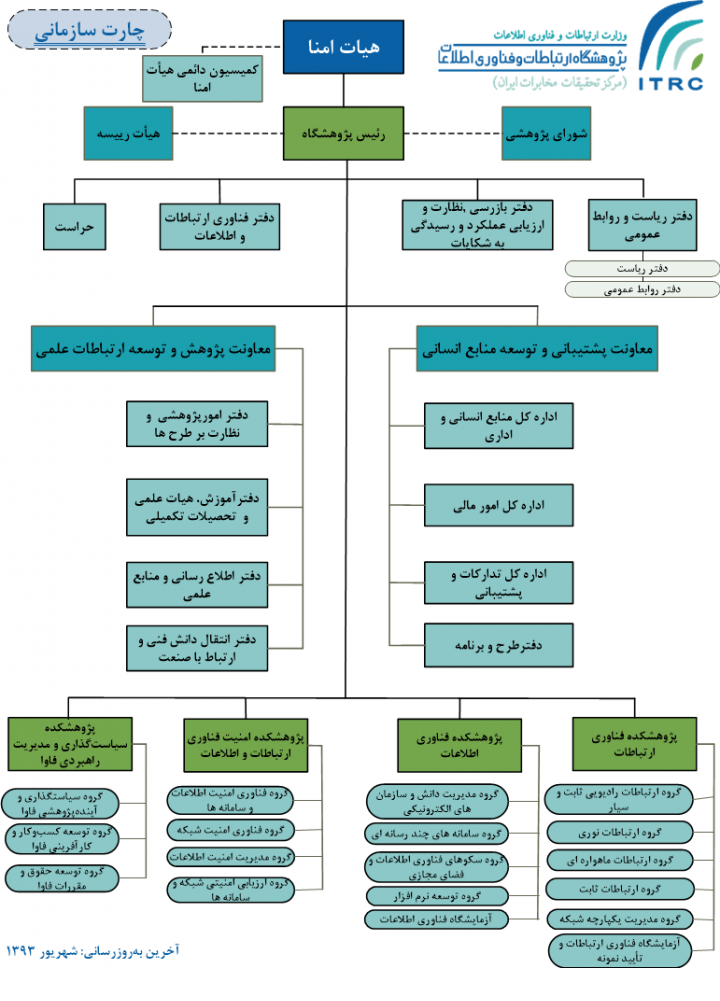
# فصل اول: مرکز تحقیقات مخابرات ایران

## ماموریت‌های سازمان

پژوهشگاه علاوه بر فعالیت‌های موثر برای رسیدن به چشم‌انداز تعیین‌شده، نقش مشاور مادر برای وزارتخانه، شرکت‌های تابعه و دیگر نهادهای حکومتی و به تبع آن نقش هماهنگ‌کننده عرصه تحقیقات علمی، فنی و اجتماعی را نیز در حوزه‌های مربوط به فناوری اطلاعات و ارتباطات (ICT یا فاوا) ایفا می‌کند. بر اساس ماموریت‌های وزارت ارتباطات در قانون برنامه‌ پنجم توسعه، اساسنامه و حکم مقام عالی وزارت در شروع دوره کاری جدید پژوهشگاه و در چارچوب اساسنامه پژوهشگاه، عمده ماموریت‌های لازم الاجرا برای پژوهشگاه عبارتند از:

* برنامه‌ریزی و حمایت منسجم جهت نهادینه شدن پژوهش حوزه فاوا در سطح کشور
* مشاوره جهت پاسخگویی به نیازهای پژوهشی سازمان‌ها و شرکت‌های تابعه وزارتخانه و سایر بخش‌های کشور در حوزه فاوا
* برنامه‌ریزی و اقدام جهت ارتقاء کیفی و کمی فناوری‌های بومی حوزه فاوا و استفاده از حداکثر توان مهندسی داخلی
* ترسیم گستره‌های تحقیقاتی جدید در حوزه فاوا مطابق مرزهای دانش و ارتقای توانمندی‌های مؤسسه از طریق مدیریت تحقیق، برونسپاری و درونسپاری فعالیت¬های پژوهشی
* برنامه ریزی برای محرومیت زدایی، عدالت اجتماعی و اقتصادی و توسعه متوازن در اجرای طرحهای تحقیقاتی و تهیه پیوستهای مختلف اقتصادی، مقرراتی، فرهنگی
* تهیه استانداردهای ملی و ارائه راهکارها و ابزار مناسب برای نظارت و ارزیابی بر استانداردها (نظیر آزمایشگاه‌های تأیید نمونه تجهیزات و خدمات)
* جلب مشارکت و تقویت حضور بخش خصوصی و تعاونی در تجاری‌سازی نتایج پژوهش و شبکه‌سازی ظرفیت‌های موجود
* زمینه‌سازی جهت خلق فرصت‌های شغلی متعدد و سازنده برای فارغ‌التحصیلان و نخبگان حوزه‌ فاوا
* برقراری تعاملات سازنده و دوسویه با پژوهشگاه‌های مشابه در کشورهای منطقه و جهان برای انتقال دانش و افزایش قدرت نرم کشور در زمینه‌های پژوهشی با حضور موثر در رویدادهای مهم علمی و تخصصی بین‌المللی
* جلب همکاری و بسترسازی برای حضور نخبگان و فرهیختگان ایرانی حوزه فاوا در خارج از کشور

## چارت سازمانی



شکل 1 چارت سازمانی پژوشگاه ارتباطات و فناوری اطلاعات

# فصل دوم: مقدمه

هنگامی که کاربران در ساختمان‌های اداری، مسکونی و تجاری توانستند از طریق اتصالات اینترنت باسیم با کاربران دیگر ارتباط برقرار کنند انقلابی در زمینه ارتباطات رخ داد. موج دوم این انقلاب زمانی بود که تجهیزات سیار مصرف کنندگان (لپ‌تاپ‌ها، گوشی‌های تلفن همراه هوشمند، تبلت‌ها) از طریق اتصالات بی‌سیم به یکدیگر و شبکه‌های عمومی متصل شدند؛ اما موج آخر مربوط به اتصال اشیا به کاربران، شبکه‌های تجاری و عمومی و دیگر اشیا از طریق ترکیبی از اتصال به اینترنت است. این موج اتومبیل‌ها، هواپیما‌ها، تجهیزات پزشکی، آسیاب‌های بادی، حسگر‌های محیطی، تجهیزات استخراج گاز طبیعی و بسیاری دیگر از دستگاه‌ها و تجهیزات را در بر می‌گیرد. کارایی و بهره‌وری این سیستم‌ها به شکل قابل توجهی به کمک اتصالات نظیر به نظیر و مشتری-سرویس‌دهنده که محصول پیشرفت‌های جدید در اتصالات و کنترل کننده‌های ارزان قیمت و پروتکل‌های استاندارد اینترنت است، افزایش می‌یابد.

اینترنت اشیا (IoT)، به اتصال فزاینده اشیا از هر نوع، از کاربردهای خانگی تا تجهیزات به کار رفته در کاربردهای صنعتی، به اینترنت یا ساختاری شبیه به اینترنت اشاره دارد. ایده‌ی اصلی در این مفهوم این است که تجهیزات هوشمند باید بتوانند با یکدیگر و با واسط‌های انسانی در سراسر جهان برای افزایش بهره‌وری ارتباط برقرار کنند.

به عقیده‌ی برخی کارشناسان، IoT پس از اینترنت و شبکه‌های تلفن همراه، سومین موج از فناوری‌های ICT خواهد بود. انستیتو فرستر پیش‌بینی کرده است که ارتباطات اینترنتی شی با شی تا سال ۲۰۲۰ در حدود ۳۰ برابر تعاملات ارتباطی انسان با انسان خواهد بود و این بدان معناست که میلیارد‌ها اتصال اینترنتی از آینده مخابرات، فقط مربوط به تعاملات بین اشیا خواهد بود.

انقلاب اینترنت منجر به ارتباط متقابل مردم با سرعت بی‌سابقه‌ای شده است. انقلاب آینده ارتباط میان اشیا برای ایجاد محیطی هوشمند خواهد بود. در سال ۲۰۱۱ تعداد دستگاه‌های متصل به یکدیگر به بیش از تعداد انسان‌های کره زمین رسید. بر اساس گزارش CISCO در سال ۲۰۱۲ حدود ۸.۷ میلیارد شی متصل به اینترنت جهانی وجود داشت. در سال ۲۰۱۳ این مقدار به بالغ بر ۱۰ میلیارد رسید. با کاهش هزینه برای هر اتصال و در نتیجه رشد سریع تعداد ارتباطات ماشین به ماشین، انتطار می‌رود که این تعداد در سال ۲۰۲۰ به ۵۰ میلیارد برسد. پیش‌بینی می‌شود زمان مورد نیاز به منظور بلوغ اینترنت اشیا در صنعت بین ۵ تا ۱۰ سال است.

راه اندازی کارخانه‌های هوشمند سال‌های قبل نیازمند سیستم‌های اتوماسیون گران‌قیمت و سفارشی بود که تنها کارخانه‌های بزرگ از پس هزینه‌های آن برمی‌آمدند. شرایط در هوشمندسازی منازل نیز به همین ترتیب بود و تنها بیلیونرهایی مانند بیل گیتس از پس هزینه‌های سیستم‌های سفارشی هوشمندسازی منازل بر‌می‌آمدند. موارد زیر را می‌توان از عوامل گسترش اینترنت اشیا در سال‌های اخیر برشمرد:

* نصف شدن قیمت سنسور‌ها و عملگرها در ۱۰ سال اخیر
* کوچکتر قدرتمندتر شدن سنسور‌ها و عملگر‌ها (سخت‌افزارها)
* دسترسی به ابزارات پشتیبانی مانند زیرساخت‌های ابری و ابزارات داده‌های عظیم

## پیشینه تحقیقاتی

اصطلاح اینترنت اشیا برای اولین بار توسط کلوین اشتون[[1]](#footnote-1) در سال 1998 بیان گردید، او بیان کرد که اینترنت اشیا پتانسیل تغییر جهان را به مانند اینترنت و حتی بیشتر از آن دارد. پس از آن مرکز MIT Auto-ID چشم انداز IoT خود را در سال 2001 ارائه کرد، ITU[[2]](#footnote-2) نیز به صورت رسمی اینترنت اشیا را در گزارش اینترنت ITU در سال 2005 معرفی کرد. تحقیقات زیادی در سراسر جهان در زمینه IoT در حال انجام است ولی این تحقیقات در حال حاضر، در مراحل ابتدایی خود قرار دارند. در زمینه تعریف «اینترنت اشیا» تعاریف مختلفی توسط محققان بیان شده است ولی هنوز تعریف جامعی از IoTبیان نشده است. در ادامه مهم‌ترین تعاریف بیان شده توسط محققان آورده شده است:

اشیاای که دارای هویت و شخصیت مجازی هستند و در فضاهای هوشمند با استفاده از رابط‌های هوشمند با اجتماع، محیط و کاربران مفهومی[[3]](#footnote-3) اتصال و ارتباط برقرار می‌کنند.

منشا معنایی اینترنت اشیا متشکل از دو کلمه و مفهوم، اینترنت و شی است که اینترنت می‌تواند به صورت یک شبکه جهانی از کامپیوترهای متصل به هم مبتنی بر پروتکل‌های ارتباطی استاندارد مانند TCP/IP تعریف شود درحالی‌که شی را نمی‌توان به صورت دقیق تعریف کرد؛ بنابراین اینترنت اشیا به طور مفهومی به معنای شبکه جهانی از اشیا با آدرس‌های منحصر به فرد به هم پیوسته بر مبنای پروتکل‌های استاندارد ارتباطی است.

اینترنت اشیا به افراد و اشیا اجازه می‌دهد که در هر زمان، در هر مکان با هر چیز و هرکس، به‌وسیله هر مسیر/ شبکه‌ای و هر سرویسی متصل بشوند.

اینترنت اشیا، الگویی است که در آن اشیا با قابلیت‌های شناسایی شدن، اندازه‌گیری، اتصال به شبکه و پردازش مجهز شده‌اند تا بتوانند به منظور انجام یک هدف خاص با یکدیگر ارتباط برقرار بکنند. هوشمندی فراگیر به اشیا، این امکان را می‌دهد که نسبت به محیط خودآگاهی پیدا کنند، با انسان‌ها تقابل کنند و تصمیم‌سازی کنند.

گسترش اشیا هوشمند در سطح جهانی می‌تواند موجب پیشرفت‌های گسترده در روندهای تجاری و زندگی انسان‌ها شود ولی در عین حال با خود تهدیدات و چالش‌های فنی بسیاری به همراه خواهد آور. همان­طور که اشاره شد، تا کنون مقاله‌های بسیاری در زمینه اینترنت اشیا منتشر شده است. می‌توان مقاله‌های منتشر شده در حوزه IoT بر اساس تمرکز مطالعاتی آن‌ها، به پنج دسته زیر تقسیم‌بندی شده است:

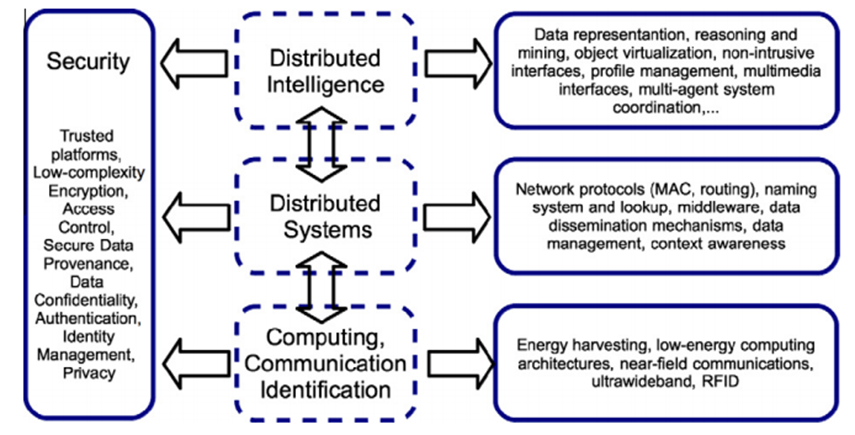
* فناوری‌های مرتبط با IoT
* کاربردهای IoT
* چالش‌ها
* مدل تجاری IoT
* مطالعه مسیرهای پیشرو در این حوزه و بررسی‌های کلی و مروری

در ادامه می‌توان موضوعات مورد مطالعه در بخش فناوری را به سه دسته سخت‌افزار، نرم‌افزار و معماری تقسیم کرده است که معماری‌ها باید ساختار و آرایش IoT را به نحوی فراهم کند که عملکرد مورد انتظار آن را به طور کارآمد تضمین کند. به منظور تضمین محیط‌های محاسباتی توزیع‌شده که از نیازمندی‌های پیاده‌سازی IoT است معماری‌هایی پیشنهاد شده است که بر مبنای ارتباط نظیر به نظیر[[4]](#footnote-4) و خودکار بیان شده­اند. نکته قابل توجه این است که تنوع معماری‌های مختلفی که برای IoT پیشنهاد شده­اند، اهمیت استانداردسازی معماری را پررنگ‌تر می‌کند.

لفظ اینترنت اشیا یک کلیدواژه کلی برای پوشش جنبه‌های مختلف از گسترش اینترنت و وب در دنیای فیزیکی است. اجزای اصلی مفهوم اینترنت اشیا را مبتنی بر قابلیت قابل توجه محاسباتی و ارتباطی، قابلیت شناسایی شدن، قابلیت اندازه‌گیری و اجرای دستورات که به صورت نهفته باید در اشیا قرار گیرند، می‌دانند. از این‌رو چالش‌های تحقیقاتی در این حوزه را به دسته‌های زیر تقسیم‌بندی کرده‌اند:

* فنّاوری‌های مربوط به قابلیت شناسایی، ارتباطی و محاسباتی اشیا
* فنّاوری سیستم‌های توزیع‌شده
* هوشمندی توزیع‌شده
* امنیت
* کاربردها و حوزه‌های تحت تاثیر
* تلاش‌های مرتبط با استاندارد‌سازی

شکل 2 نمایی از چالش‌های تحقیقاتی در IoT را بیان می‌کند.



شکل 2 تقسیم‌بندی حوزه‌های تحقیقاتی مرتبط با اینترنت اشیا

# فصل سوم: شهرهای هوشمند

## مقدمه

شهر‌ هوشمند یک نگاه توسعه شهری برای به کارگیری روش‌هایی از اینترنت اشیا (IoT) و ICT به صورت امن برای مدیریت دارایی‌های شهری می‌باشد. هدف شهر‌های هوشمند افزایش کیفیت زندگی با استفاده از انفورماتیک شهری و تکنولوژی می‌باشد که به این ترتیب کیفیت سرویس‌ها افزایش می‌یابد.



شکل 3 شمای یک شهر هوشمند

از جمله کاربرد‌های اینترنت اشیا در شهر هوشمند می‌توان به مدیریت ترافیک، نور معابر، سیستم‌های سلامت شهری و سیستم‌های نظارت شهری اشاره کرد.

یک دسته‌ی بزرگ از شرکت‌های هوشمندسازی شهر‌ها در شهر تل‌‌آویو در اسرائیل قرار دارد و این شهر جایزه‌ی جهانی شهر هوشمند را در سال ۲۰۱۴ برنده شد، شرکت‌های اسرائیلی این راه‌کارها را به صورت جهانی پیاده‌سازی می‌کنند.

## ویژگی‌ها

شهرهای هوشمند به صورت مشخص تعریف نشده‌اند ولی یک شهر هوشمند به طور کلی از فناوری اطلاعات استفاده می‌کند تا بتواند:

* با استفاده از آنالیز داده‌ای و هوش مصنوعی از بستر فیزیکی (مانند جاده‌ها، دارایی‌های فیزیکی و ..) بهینه‌تر استفاده کند.

این فرم‌ها از هوشمندی در شهر‌ها می‌تواند به سه صورت‌ زیر نمود پیدا کند:

* هماهنگی‌های هوشمند: شهرهای هوشمند همکاری و ارتباطات بین انجمن‌ها و موسسات برای حل مشکلات را فراهم می‌آورد.
* اختیارات هوشمند: شهرها با فراهم آوردن بستر‌های باز و امکانات آزمایشی اجازه‌ی خلاقیت و نوآوری در مناطق را می‌دهند.
* ابزار‌های هوشمند: با وجود اینکه زیرساخت شهری با استفاده از جمع‌آوری داده‌ی بلادرنگ و پردازش و مدل‌های پیش‌بینی هوشمند می‌گردد. بحث‌های زیادی در این مورد وجود دارد که از جمله‌ی آن مشکلات نظارتی در شهر‌های هوشمند می‌باشد.

## چالش‌ها

به طور کلی چالش‌های زیر برای اینترنت اشیا مطرح هستند که قطعا در زمینه‌ی شهر‌های هوشمند نیز وجود دارند:

* تعداد بسیار زیاد اشیا

سیر کنونی در افزایش تعداد دستگاه‌های هوشمند نشان‌دهنده اتصال هزاران شی به اینترنت در آینده نه چندان دور است. چگونگی تایید هویت در دسترسی، نگهداری، محافظت، استفاده و پشتیبانی از این مقیاس بزرگ اشیا به عنوان یک چالش اساسی مطرح است. مهم‌ترین سوالاتی که باید در این زمینه پاسخ داده شوند عبارت‌اند از:

* آیا پشتیبانی از پروتکل IPv6 کافی خواهد بود؟
* آیا پروتکل‌های دیگری مانند IPv6 در آینده اینترنت اشیا حضور خواهند داشت؟
* آیا پروتکل‌ها و استانداردهای جدیدی به وجود خواهد آمد؟
* چگونه حجم زیاد اطلاعات جمع‌آوری، ذخیره و استفاده خواهد شد؟

• تولید دانش و داده بزرگ

در دنیای IoTداده‌های خام بسیاری وجود دارند که پیوسته جمع‌آوری شده‌اند. بنابراین تکنیک‌هایی لازم است که این داده‌های خام را به دانش کاربردی مبدل سازند. چالش اصلی در تفسیر داده‌ها و شکل‌گیری دانش، آدرس‌دهی اطلاعات جهان فیزیکی و توسعه روش‌های استنتاج جدیدی است که محدودیت‌های روش‌های سنتی را نداشته باشد. در نظر گرفتن هزینه محاسبات از جمله این محدودیت‌ها است.

مقدار زیادی داده از حسگرهای دستگاه‌ها در اینترنت اشیا جمع‌آوری خواهد شد که با توجه به اهداف مختلف و به شیوه‌های مختلفی مورد استفاده قرار خواهد گرفت. تکنیک‌های داده‌کاوی باید با استفاده از این داده‌ها، دانش جدیدی را خلق کنند. یکی از مشکلات پژوهشی استفاده از این جریان داده‌ها به عنوان داده اصلی برای استنتاج‌های غیرمنتظره آینده است. هیچ روش استنتاجی بدون خطا نیست و ممکن است کاربران به علت وجود عدم قطعیت در داده‌های تفسیر شده، به سیستم بی‌اعتماد شوند. اعتماد یکی از جنبه‌های پر اهمیت در سودمندی داده بزرگ است و دارای عناصر اصلی امنیت و حریم خصوصی است. به این ترتیب باید روش‌های جدیدی برای کالیبراسیون حسگرهای موجود و همچنین پروتکل‌ها در این زمینه به منظور قابلیت اطمینان انجام گیرد. بدون این عوامل، استنتاج ممکن است بر اساس داده‌های اشتباه و یا کمی انجام گیرد که خود می‌تواند باعث بروز مشکلات جدیدی شود.

یکی دیگر از چالش‌های موجود، تصمیم‌گیری درست بر مبنای دانش به وجود آمده است. در تصمیم‌گیری لازم است موارد منفی و مثبت کاذب حداقل شده و ایمنی تصمیم تضمین شود زیرا در غیر این صورت سیستم غیرقابل اعتماد خواهد بود.

بسیاری از برنامه‌های کاربردی شخصی‌سازی ‌شده و در آینده برای فرد خاصی طراحی خواهند شد. به این ترتیب باید تضمین شود که اطلاعات جمع‌آوری شده و عواقب آن مرتبط با فرد مورد نظر است. این مسئله نیز چالش برانگیز خواهد بود اما تا حدودی، بدون در نظر گرفتن مسائل حریم خصوصی، با استفاده کاربران از ابزار قابل پوشش RFID ها و یا دوربین‌های دارای تشخیص الگو این مسئله حل شده است. با این وجود در موارد دیگر لازم است از مجموعه‌ای از حسگرهای دارای قابلیت ردیابی و همچنین تاریخچه‌ای از فعالیت‌های کاربران و مشخصات فردی آن‌ها استفاده شود تا بتوان به تخصیص داده دقیقی دست یافت. به این ترتیب پژوهش در این زمینه ضروری خواهد بود.

• مقاوم بودن و انعطاف پذیری

بسیاری از برنامه‌های کاربردی IoT مبتنی بر پلتفرم ارتباطی (اتصال‌دهنده شبکه‌ای از اشیا)، حسگر و عملگر خواهند بود. در این نوع ساختار، دستگاه‌ها به طور معمول مکان‌ها را می‌دانند، کلاک های همگام‌سازی شده دارند، به هنگام همکاری از دستگاه‌ها در همسایگی خود اطلاع داشته، دارای مجموعه‌ای منسجم از پارامترهای تنظیم‌شده نظیر زمان‌بندی بیداری بوده و کلیدهای امنیتی مورد نیاز برای تبادل اطلاعات با یکدیگر را دارند. با این حال با گذشت زمان برخی از این شرایط رو به زوال خواهد رفت. از آن جمله می‌توان همگام‌سازی کلاک را نام برد. رانش کلاک باعث می‌شود که گره‌ها زمان‌های مختلفی داشته باشند که باعث ایجاد اختلال در برنامه کاربردی می‌گردد. همچنین کنترل عملگرها به دلیل پروتکل‌ها و نرم‌افزارهای کنترلی آن‌ها و سایش و گسستگی فیزیکی ممکن است با خرابی مواجه شود. به ‌این ‌ترتیب چگونه یک IoT متحرک، پویا و درازمدت می‌تواند حفظ شود؟

خدمات منسجم مورد نظر باید با بسیاری از روش‌های دیگر ترکیب شود تا عملکرد قابل اطمینانی را برای سیستم ایجاد کند. هدف به دست آوردن مجموعه راه‌حل‌هایی جهت ایجاد سیستمی قابل‌اطمینان با توجه به وجود نویز، خرابی و عدم قطعیت در دنیای فیزیکی است.

مشکل دیگر در برخی از برنامه‌های کاربردی IoT از جمله انواع حیاتی و ایمنی آن این است که تضمینی از زمان اجرا باید به مقامات مسئول ارائه گردد؛ بنابراین چنین برنامه‌های کاربردی IoT نیاز به خدماتی دارند که از صدور گواهینامه‌ای برای زمان اجرا پشتیبانی کند.

• باز بودن

بسیاری از سیستم‌هایی که مبتنی بر حسگر هستند، سیستم‌های بسته‌ای هستند. همکاری و تعامل چنین اشیایی با یکدیگر در حال افزایش است و برای رسیدن به مزیت‌های این تعامل باید شی مورد نظر دارای یک ساختار باز باشد. پشتیبانی از باز بودن سیستم‌ها مسائل پژوهشی جدیدی را به وجود می‌آورد. همچنین به منظور کارآمدی تبادل اطلاعات در سراسر سیستم‌های متنوع، به رابط‌های ارتباطاتی یکپارچه و جدیدی نیاز است. باید توجه داشت که باز بودن سیستم نیز می‌تواند مشکلاتی را در زمینه امنیت و حریم خصوصی به وجود آورد و نیاز است تعادل صحیحی میان دسترسی به عملکرد و امنیت و حفظ حریم خصوصی در سیستم باز فراهم شود.

بسیاری از سیستم‌های حسگر و عملگر به منظور عملکرد مقاوم از کنترل بازخورد استفاده می‌کنند. در روش‌های سنتی، مدل سیستم ایجادشده و سپس برای آن کنترل‌کننده طراحی می‌گردد اما در سیستم‌های باز مدل سیستم به طور مداوم تغییر کرده و همچنین تعاملات انسانی با آن‌ها جز جدایی‌ناپذیر این سیستم‌ها است که فرآیند مدل‌سازی را بسیار دشوار می‌سازد. اگرچه روش‌هایی نظیر کنترل تصادفی، کنترل مقاوم و کنترل تطبیقی توسعه داده ‌شده‌اند اما برای پشتیبانی از درجه باز بودن و پویایی برخی از سیستم‌های IoT کفایت نمی‌کنند و به مجموعه‌ای از تکنیک‌ها و نظریه‌های جدیدتر و غنی‌تری در این حوزه نیاز است. وابستگی‌های میان حلقه‌های کنترلی باید به روش تطبیقی و در زمان حقیقی مورد توجه قرار گیرد تا بتواند از باز بودن سیستم‌ها در IoT حمایت کند.

• امنیت

مشکل اساسی که امروزه در اینترنت فراگیر شده است، برخورد با حملات امنیتی است. حملات امنیتی به دلایل منابع محدود اشیا، دسترسی فیزیکی به حسگرها، عملگرها و اشیا، باز بودن سیستم‌ها و ارتباطات بی‌سیم میان بسیاری از دستگاه‌ها، در حوزه‌ IoT مشکل ساز هستند.

افزونگی این امکان را مهیا می‌سازد که با وجود خرابی همچنان برنامه‌های کاربردی به ارائه خدمات خاص خود ادامه دهند. برنامه‌های کاربردی IoT باید بتوانند در حضور حملات امنیتی به صورت رضایت بخشی به فعالیت‌های خود ادامه دهند. امروزه بسیاری از راه‌حل‌های امنیتی نیازمند محاسبات هستند و یکی از چالش‌های پژوهشی اصلی برای IoT به حساب می‌آیند. همچنین پشتیبانی قابل توجهی از سخت‌افزار جهت رمزنگاری، احراز اصالت و تصدیق امضا مورد نیاز است.

• حریم خصوصی

فراگیری و تعاملات موجود در IoT برای افراد خدمات مفید و کارآمدی را به ارمغان آورده است اما فرصت‌هایی را نیز ایجاد کرده است که منجر به نقض حریم خصوصی آن‌ها خواهد شد. به منظور حل این مشکل در برنامه‌های کاربردی IoT که در آینده به وجود می‌آید، باید سیاست‌های حفظ حریم خصوصی در هر سیستمی مشخص شود. بدین ترتیب الگوی IoT قادر خواهد بود که در اجازه دسترسی به کاربران در دستیابی به اطلاعات تصمیم‌گیری کند. در راستای ارائه سیاست‌های حفظ حریم خصوصی به زیان جدیدی نیاز خواهد بود زیرا:

نیاز است انواع مختلف محتواهای محیط از جمله زمان، مکان، سنجش فیزیولوژیکی و سنجش محیطی بیان شود. بسیاری از این محتواها نیاز دارند که بی‌درنگ جمع‌آوری و ارزیابی شوند.

نیاز است انواع مختلفی از دارندگان داده ارائه شود. برخلاف سیستم‌های معمول حفظ حریم شخصی که در آن دارندگان داده افراد هستند، زبان حفظ حریم خصوصی IoT باید از اشیای فیزیکی مانند یخچال و فریزر نیز به عنوان دارندگان داده حمایت کند.

نیاز است که درخواست‌های جمع‌آوری سطح بالایی مانند حداکثر و حداقل خواندن داده‌های سنجش خاصی، ارائه شود. این قابلیت باید با استفاده از توابع تجمعی ناشناس سازی حمایت شود و لازم است برای جریان‌های زمان حقیقی و در سراسر منابع داده‌های بزرگ وجود داشته باشد.

نه تنها به پشتیبانی و پایبندی به حریم خصوصی در نمایش اطلاعات نیاز است بلکه باید در صورت تقاضا بر روی پارامترهای مورد نظر سیستم نیز حریم خصوصی گذاشته شود؛ به عنوان مثال استفاده خصوصی از عملگرها.

باید اجازه تغییرات پویا در سیاست‌ها صادر شود. یکی از مشکلات، تعامل سیستم‌ها با سیستم‌های دیگر است و هر یک سیاست‌های حفظ حریم خصوصی خود را دارند؛ بنابراین در دنیای IoT ناسازگاری و تناقض میان سیستم‌ها رخ خواهد داد و به بررسی سازگاری برخط، اطلاع‌رسانی و سیاست‌های بدون ابهام نیاز است.

• حضور انسان‌ها در حلقه

گسترش کاربردهای اینترنت اشیا سبب پیچیدگی برنامه‌های کاربردی آن خواهد شد. بسیاری از این برنامه‌های جدید در ارتباط تنگاتنگی با انسان‌ها خواهند بود. حضور انسان در حلقه سیستم‌ها، فرصت‌های جدیدی را در طیف گسترده‌ای از برنامه‌های کاربردی از جمله مدیریت انرژی به وجود می‌آورد و می‌توان از الگوی زندگی بشر در طراحی این برنامه‌های کاربردی سود برد؛ اما مدل‌سازی رفتار انسان به دلیل جنبه‌های پیچیده فیزیولوژیکی، روانی و رفتاری بشر بسیار چالش برانگیز است. ازاین‌رو به پژوهش‌های جدیدی جهت رساندن انسان به یک محور مرکزی در طراحی سیستم‌ها و همچنین حل سه چالش اصلی زیر نیاز است:

* نیاز به درک مفهومی از طیف کلی انواع کنترل‌های انسان موجود در حلقه کنترل
* نیاز به‌ ضمیمه‌هایی برای شناسایی سیستم و یا روش‌های دیگری به منظور مدل‌سازی رفتار انسان
* تعیین چگونگی ترکیب مدل‌های رفتار انسانی با روش مرسوم کنترل بازخورد

• دسترسی‌پذیری

دسترسی‌پذیری باید به گونه در سخت‌افزار و نرم‌افزار تحقق یابد که امکان دسترسی در همه‌جا و هرزمانی را برای مشتریان فراهم کند. دسترسی‌پذیری برای نرم‌افزار به صورت امکان فراهم کردن سرویس برای مشتریان در مکان‌های متفاوت و به صورت هم‌زمان تعریف می‌شود. دسترسی‌پذیری برای سخت‌افزار به معنی وجود داشتن سخت‌افزارهایی سازگار با کاربردها و پروتکل‌های IoT در هرزمانی است. یک روش برای فراهم آوردن دسترسی‌پذیری می‌تواند ساخت دستگاه‌ها و سرویس‌های پشتیبان برای دستگاه‌ها و سرویس‌های موجود باشد.

• پایداری

پایداری به درست کارکردن سیستم بر اساس مشخصات از پیش تعریف شده گفته می‌شود. در IoT به پایداری برای افزایش نرخ تحویل سرویس نیاز است. در حقیقت با پایداری دسترسی‌پذیری سیستم در طول زمان تضمین می‌شود. در کاربردهای حیاتی مانند سیستم‌های کنترل سلامت، پایداری بسیار مهم است. پایداری در تمام لایه‌های پیاده‌سازی IoT باید در نظر گرفته شود. به عنوان مثال در صورتی که ارتباطات ناپایدار باشند و بسته‌ها حذف شوند ممکن است اطلاعات حیاتی از بین بروند که منجر به تصمیمات نادرستی شده و هدف سیستم را نقض کنند. به همین منظور ارتباطات قابل‌اطمینان و مکانیسم‌های برای جلوگیری از حذف بسته در کاربردهای حیاتی باید در نظر گرفته شود.

• تحرک

بسیاری از گره‌های IoT متحرک و یا سرویس به کاربران متحرک داده می‌شود. در زمان ارائه سرویس، تحرک گره و یا کاربر نباید باعث اختلال در ارائه سرویس شود. اختلال در سرویس ممکن است به خاطر جدا شدن از یک شبکه و اتصال به شبکه دیگری رخ دهد. به همین منظور مکانیسم‌های مدیریت تحرک گره‌ها و کاربران، مانند روش‌های ارائه‌شده در شبکه‌های سلولی، در نظر گرفته شود.

• کارایی

در کارایی سرویس‌های ارائه‌شده توسط IoT، کارایی لایه‌های پایین پشته شبکه از جمله لایه فیزیکی و پیوند داده نیز باید در نظر گرفته شود. معیارهای ارزیابی کارایی نیز با توجه به کاربردها متفاوت است. انرژی مصرفی، نرخ گذردهی، تاخیر انتها به انتها و هزینه پیاده‌سازی از مهم‌ترین معیارهای کارایی سرویس‌های IoT هستند. با توجه به گستردگی تکنولوژی‌های استفاده شده، نیاز به مقایسه و ارزیابی کارایی راه‌حل‌های مختلف و بهبود کارایی آن‌ها احساس می‌شود.

• مدیریت

جنبه‌های مدیریتی هزاران دستگاه که با یکدیگر ارتباط دارند چالش بسیار مهمی برای فراهم‌کنندگان سرویس است. بسیاری از این دستگاه‌ها منابع انرژی و محاسباتی محدودی دارند و نیازمند پروتکل‌های مدیریتی جدید با سربار پردازشی و ارتباطی بسیار کم هستند. به‌ عنوان مثال یکی از جنبه‌های مهم مدیریتی نظارت بر اتصالات برقرار شده بین گره‌ها است که باید با توجه به منابع محدود گره‌ها انجام شود تا مانعی در سرویس‌های ارائه شده توسط گره‌ها ایجاد نکند.

• سازگاری

سازگاری در دستگاه‌های IoT که هرکدام ممکن است تکنولوژی‌های مختلفی را در لایه‌های مختلفی پیاده کرده باشند اهمیت فراوانی دارد. این دستگاه‌ها با وجود تفاوت‌های تکنولوژی باید قادر به تعامل و ارتباط با یکدیگر باشند. تعامل باید توسط توسعه‌دهندگان کاربردها و سرویس‌ها و سازندگان سخت‌افزارها در نظر گرفته شود تا فارغ از نوع سخت‌افزار و نرم‌افزار استفاده شده، قادر به ارائه سرویس یکپارچه به مشتریان شوند. در این بین نقش سازمان استانداردسازی و پیاده‌سازی استانداردهای مختلف نیز نباید نادیده گرفته شود. گستردگی سازمان‌های استاندارد گذاری، تنوع استانداردها و نبود استاندارد غالب در این زمینه از دیگر چالش‌های مطرح در زمینه IoT است.

## نمونه‌های بارز

* آمستردام: بنیان‌های شهر هوشمند آمستردام در سال ۲۰۰۹ آغاز شد که اکنون شامل ۷۹ پروژه می‌باشد که با همکاری تجارت‌ها، دولت و ساکنان محلی در حال توسعه‌اند. این پروژه‌ها روی یک بستر به هم پبوسته از طریق دستگاه‌های بی‌سیم اجرا می‌شوند تا توانایی شهر در تصمیم‌های آنی افزایش یابد. شهر آمستردام هدف خود را از هوشمندسازی کاهش ترافیک، حفظ انرژی و افزایش امنیت عمومی بیان داشته است.



شکل 4 شهرهوشمند آمستردام

* بارسلونا: بارسلونا تعدادی پروژه را کلید زده است که می‌توانند برنامه‌های شهر هوشمند با استراتژی سیستم عامل شهر (CityOS) تلقی گردند. بارسلونا یک شبکه جدید اتوبوس نیز بر اساس آنالیز داده‌ای از جریان‌های ترافیک معمول در بارسلونا طراحی کرده است.



شکل 5 شبکه‌ی جدید اتوبوس در بارسلونا

به دلیل کاهش روز افزون منابع آب‌های زیرزمینی به خصوص در کشورهایی که با بحران آب مواجه هستند، جمع آوری اطلاعات کنتورهای هوشمند که می‌توان آن را از زیر کاربردهای شهر هوشمند در نظر گرفت، بسیار مورد توجه قرار گرفته است.

در کشور عربستان و در شهرهای دمام، مکه، مدینه، ریاض و جده بستری به منظور جمع‌آوری داده و استخراج الگوی مصرف آب پیاده‌سازی شده است.

در کشور اسپانیا در استان‌هایی نظیر کاستلیون در راستای پروژه شهر هوشمند قسمت بسیار زیادی از شهرها تحت پوشش بستر هوشمند جمع‌آوری اطلاعات کنتورهای هوشمند آب رفته‌اند.

در کشور آمریکا و در ایالت کالیفرنیا نیز پروژه جمع‌آوری اطلاعات کنتورهای هوشمند آب به صورت اتوماتیک، به بهره‌برداری رسیده است. که این سامانه با دریافت اطلاعات هر مشترک به صورت چهار بار در روز، اطلاعات مورد نیاز را استخراج و در مواقع لزوم نیز به مشترکین اطلاع‌رسانی می‌کند.

# فصل سوم: بستر‌های اینترنت اشیا

## مقدمه

امروزه شاهد گسترش بی‌اندازه‌ی بسترهای اینترنت اشیا در جهان می‌باشیم و پیش‌بینی می‌شود که بازار اینترنت اشیا تا سال ۲۰۱۹ ارزشی بالغ بر ۱ بیلیون دلار پیدا کند.

بسترهای اینترنت اشیا کلید پیاده‌سازی برنامه‌ها و سرویس‌های مقیاس‌پذیر اینترنت اشیا می‌باشند که دنیا واقعی و مجازی میان اشیا و مردم را به یکدیگر متصل می‌کند. با وجود اینکه بازار بستر‌های اینترنت اشیا شاخه‌ی جدیدی می‌باشد که در چند ساله‌ی اخیر وجود نداشته است ولی چشم انداز آن پیچیده بوده و به سرعت تغییر می‌کند.

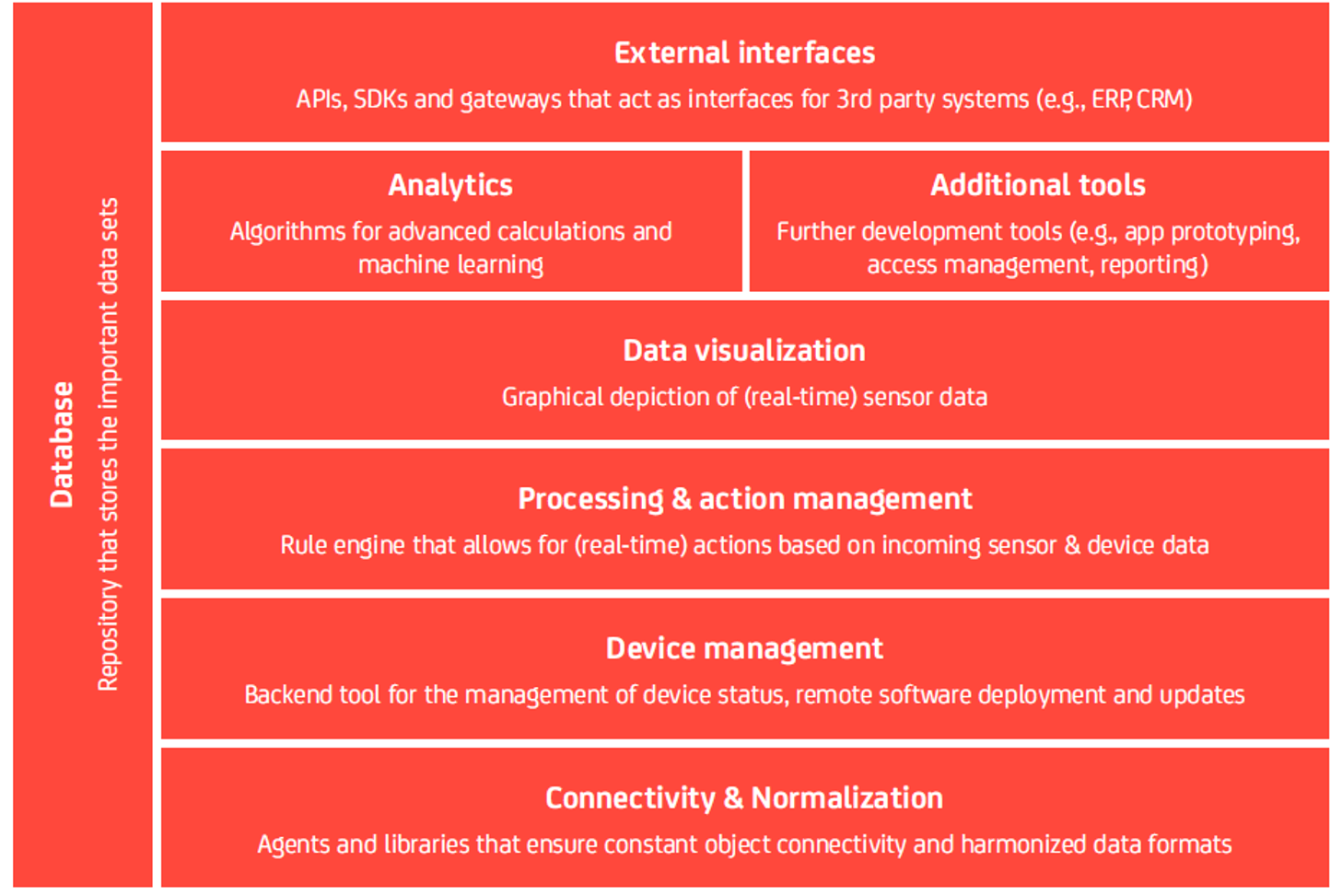
زمانی که ما در مورد بستر‌های اینترنت اشیا صحبت می‌کنیم مقصود بستر توانمند‌ساز برنامه‌های اینترنت اشیا [[5]](#footnote-5)می‌باشد.

۴ نوع دیگر از بسترها نیز وجود دارند که عموما به عنوان بستر اینترنت اشیا مطرح می‌گردند:

* بسترهای ارتباطی / M2M: این بسترها به صورت عمده بر روی ارتباط اشیا متصل از طریق شبکه‌های ارتباطی تمرکز دارند.
* پشتیبان‌های زیرساخت به عنوان سرویس[[6]](#footnote-6): این بسترها توان پردازشی و فضای ذخیره‌سازی برای برنامه‌ها و سرویس‌ها فراهم می‌آورند.
* بستر‌های نرم‌افزاری خاص سخت‌افزار[[7]](#footnote-7): بعضی از شرکت‌ها سخت‌‌افزارهای خود را با یک بستر نرم‌افزاری انحصاری به فروش می‌رسانند و به این بستر‌ها، بستر‌های اینترنت اشیا می‌گویند. از آنجایی که این بستر‌ها باز نیستند و نمی‌توان اشیا مختلفی را به آن‌ها متصل کرد، بستر اینترنت اشیا به حساب نمی‌آیند.
* الحقات نرم‌افزاری سازمانی[[8]](#footnote-8): بستر‌های نرم‌افزاری سازمانی موجود مانند Windows 10 به صورت گسترده اجازه‌ی ادغام با دستگاه‌های اینترنت اشیا را می‌دهد. در حال حاضر این الحاقات نرم‌افزاری به قدری پیشرفته نشده‌اند که یک بستر اینترنت اشیا را تشکیل دهند.

## ۸ قسمت اصلی بستر اینترنت اشیا

در ساده‌ترین حالت بستر اینترنت اشیا تنها در مورد فعال کردن ارتباط بین اشیا می‌باشد. در فرم‌های پیچیده‌تر یک بستر اینترنت اشیا شامل بلاک‌های سازنده‌ی متفاوتی ‌می‌باشد.



شکل 6 ۸ بلاک سازنده‌ی اصلی یک بستر اینترنت اشیا

### اتصال و نرمال‌سازی

هر بستر اینترنت اشیا با یک لایه اتصال آغاز می‌شود که قابلیت تبدیل پروتکل‌های مختلف و نوع‌های داده‌ای مختلف به یک رابط نرم افزاری را دارد. این لایه برای اطمینان از تقابلات درست نود‌ها با داده‌ها ضروری می‌باشد.

دستگاه‌های پیشرفته عموما یک رابط استاندارد برای ارتباط فراهم می‌آورند. با این وجود در بیشتر موارد بسترهای اینترنت اشیا عامل‌های نرم افزاری برای قرار گرفتن روی اشیا و ایجاد ارتباط فراهم می‌آورند.

### مدیریت دستگاه‌ها

واحد مدیریت دستگاه‌ها در یک بستر اینترنت اشیا اطمینان می‌دهد دستگاه‌های متصل شده به درستی کار می‌کنند و نرم‌افزارهای آن‌ها در به روز و در حال فعالیت می‌باشد.

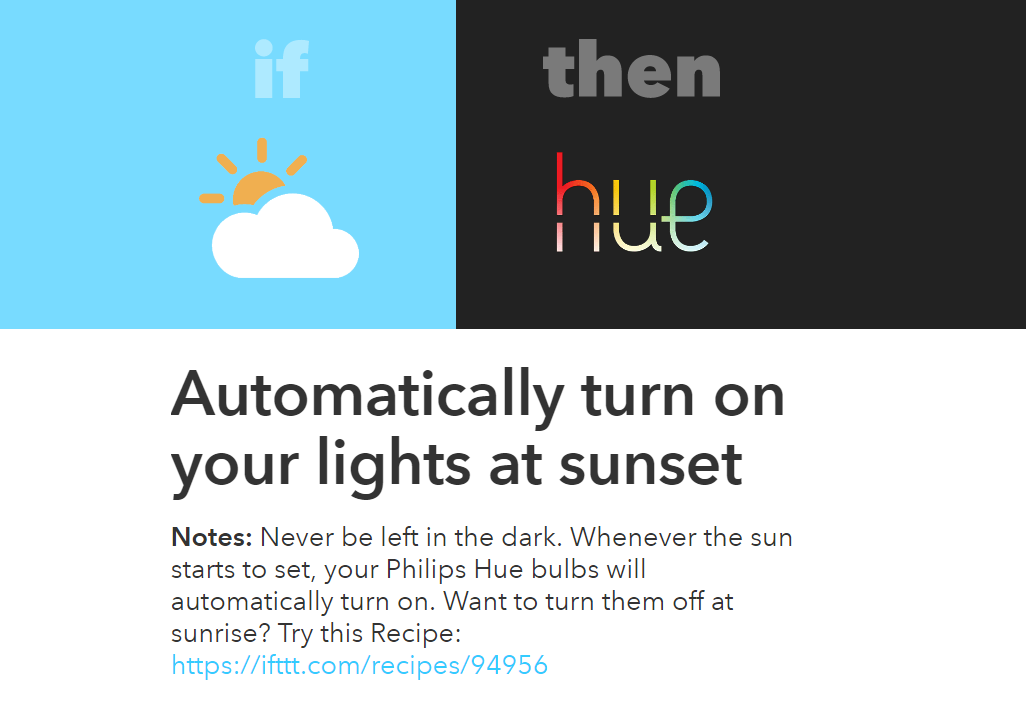
### پایگاه داده

پایگاه داده یک قسمت اصلی در یک بستر اینترنت اشیا می‌باشد. مدیریت داده‌های نودها نیازمندی‌های پایگاه داده را وارد سطح جدیدی می‌کند.

* حجم: مقدار داده‌هایی که از که نیاز است که ذخیره شوند می‌تواند بسیار زیاد باشد. در بسیاری از راه‌حل‌های اینترنت اشیا تنها بخش اندکی از داده‌های تولید شده می‌تواند ذخیره شود.
* تنوع: دستگاه‌ها و سنسورهای مختلف می‌توانند فرم‌های مختلفی از داده‌ها را تولید کنند.
* سرعت: در بسیاری از موارد در اینترنت اشیا نیاز می‌باشد که جریان داده‌ها برای تصمیمات لحظه‌ای آنالیز شود.
* صحت: در بعضی از موارد سنسورها داده‌های نادقیقی تولید می‌کنند.

### پردازش و مدیریت عمل‌ها

داده در قسمت ارتباط و نرمال‌سازی جمع و در پایگاه داده ذخیره می‌گردد. رویداد-عمل-باعث [[9]](#footnote-9)اجازه رخ دادن عمل‌ها هوشمند بر اساس داده‌های یک سنسور مشخص را می‌دهد. پیاده‌سازی واقعی عملا به صورت IF-THIS-THEN-THAT (IFTTT) انجام می‌پذیرد.



شکل 7 راه‌کار IFTTT برای روشن کردن خودکار لامپ‌ها در صبح

### تجزیه و تحلیل

بیشتر موارد استفاده اینترنت اشیا از مدیریت عمل‌ها فراتر می‌رود و نیازمند تجزیه و تحلیل پیچیده برای بیشترین استفاده از جریان داده‌ها می‌باشد. موتور تجزیه و تحلیل در یک بستر اینترنت اشیا همه‌ی پردازش‌های پویا داده از یادگیری ماشین تا خوشه‌بندی داده‌ها را می‌بایست انجام دهد.

### مجسم‌سازی داده‌ها

ترکیب چشم و معز انسان هنوز بسیار فراتر از پیشرفته‌ترین موتور‌های تجزیه تحلیل می‌باشد. به همین علت مسجم‌سازی داده‌ها در یک بستر اینترنت اشیا بسیار مهم می‌باشد و انسان‌ها را قادر می‌سازد تا الگوها را بین داده‌ها ببیند.

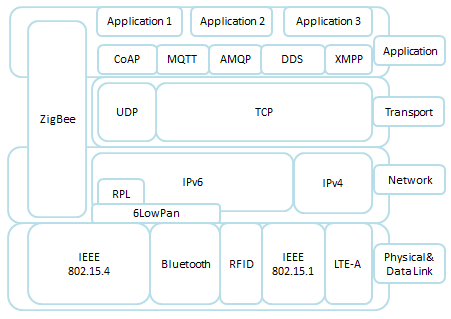
### ابزارات اضافه

بستر‌های پیشرفته اینترنت اشیا عموما مجموعه‌ای از ابزارها برای توسعه دهندگان و مدیران یک راه حل اینترنت اشیا را نیز به همراه دارند. ابزارات توسعه‌دهندگان به توسعه دهنده اجازه می‌دهد که موارد اینترنت اشیا را شبیه‌سازی و آزمایش نماید. ابزارات مدیریتی بیشتر روی کارهای روزانه اینترنت اشیا مانند مدیریت اشیا و .. تمرکز دارند.

## پروتکل‌های ارتباطی

بخش ارتباطی هر گره IoT را می‌توان به لایه‌های کاربرد، حمل، شبکه، فیزیکی و پیوند داده تقسیم‌بندی کرد. سازمان‌های استاندارد گذاری متعددی در حوزه IoT فعال هستند و هر یک استانداردها و پروتکل‌های متعددی را در لایه‌های مختلف پیشنهاد کرده‌اند که از آن جمله می‌توان IEEE، IETF، ISO و ETSI را نام برد. در این بخش به پروتکل‌های مطرح ارائه شده در زمینه ارتباطات گره‌های IoT می‌پردازیم. اگرچه نیازی به پیاده‌سازی هم‌زمان همه این پروتکل‌ها بر روی گره‌های IoT نیست و وابسته به کاربرد، می‌توان پشته پروتکلی مناسبی را انتخاب و پیاده‌سازی کرد.

پروتکل لایه شبکه پیشنهاد شده برای IoT، IPv6 است. مهم‌ترین دلیل این انتخاب استفاده از فضای آدرس گسترده آن است؛ بنابراین در بسیاری از پشته‌های شبکه پیشنهاد شده، پروتکل لایه شبکه IPv6 انتخاب شده است. اگرچه سوالاتی مانند پشتیبانی از IPv4 در همه دستگاه‌ها و یا مکانیسم تبدیل آدرس مناسب، پاسخ داده نشده باقی مانده‌اند. با توجه به ویژگی‌های IPv6 انتخاب تکنولوژی‌های لایه فیزیکی و پیوند داده نیز تحث تاثیر قرار می‌گیرد و ممکن است نیاز به یک لایه تطبیق برای انتقال بسته‌های IPv6 بر روی یک تکنولوژی خاص باشد. در **Error! Reference source not found.** پروتکل‌های ارتباطی مطرح در IoT بر حسب لایه‌های پیاده‌سازی رسم شده‌اند.



شکل 8 پروتکل‌های ارتباطی مطرح در اینترنت اشیا

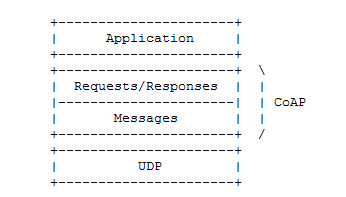
از آنجایی که در استانداردسازی‌های انجام شده، پروتکل لایه حمل جدیدی پیشنهاد نشده است بنابراین در ادامه به بررسی لایه‌های کاربرد، شبکه و لایه‌های فیزیکی و پیوند داده خواهیم پرداخت. همچنین با توجه به اینکه قسمت اعظم استاندارد ZigBee به بخش شبکه این پروتکل اختصاص دارد، بنابراین آن را در بخش شبکه مورد بررسی قرار خواهیم داد.

### پروتکل‌های لایه کاربرد

در این بخش پروتکل‌های CoAP، MQTT،AMQP ، DDS و XMPP مورد بررسی قرار گرفته‌اند. این پروتکل‌ها از معماری‌های مختلفی پشتیبانی می‌کنند و اهداف طراحی و کارایی آن‌ها با یکدیگر متفاوت است؛ بنابراین انتخاب یک پروتکل مناسب باید با در نظر گرفتن نیازمندی‌های کاربرد انجام شود.

### پروتکل CoAP

پروتکل [[10]](#footnote-10)CoAP که توسط گروه کاری IETF CoRE برای استفاده در گره‌های با منابع محدود و شبکه‌های محدود، مانند شبکه‌هایی با اتصالات متناوب و نرخ خطای بالا، پیشنهاد شده است. اهداف اصلی در طراحی این پروتکل، فراهم کردن معماری مشتری-سرویس دهنده برای استفاده از منابع، همراه با قابلیت کشف سرویس و منابع با استفاده از URI[[11]](#footnote-11)، سربار پیغام پایین و پشتیبانی از چندپخشی[[12]](#footnote-12) است. این پروتکل امکان تبادل اطلاعات مشابه با پروتکل REST را فراهم می‌سازد. REST پروتکلی برای انتقال اطلاعات در یک معماری مشتری-سرویس دهنده با استفاده از پروتکل HTTP است. سرویس دهندگان با استفاده از پروتکل REST، می‌توانند سرویس‌های خود را به آسانی در اختیار مشتریان قرار دهند. در این پروتکل از URI برای شناسایی منابع سرویس دهنده و متدهای GET، POST، PUT و DELETE پروتکل HTTP برای دسترسی به آن‌ها استفاده می‌شود. درCoAP کشف منابع با ارسال درخواست به یک آدرس شناخته شده در سرور انجام می‌شود که در پاسخ، منابعی که سرور مورد نظر ارائه می‌دهند لیست شده است. در CoAP برخلاف REST از پروتکل لایه حمل UDP استفاده می‌شود. می‌توان برای ارتباط میان CoAP و REST در محیط اینترنت از یک پروکسی برای تبدیل پیغام‌ها استفاده کرد. مطابق **Error! Reference source not found.** پروتکل CoAP از دو زیر بخش Messaging و Request/Response تشکیل شده است.

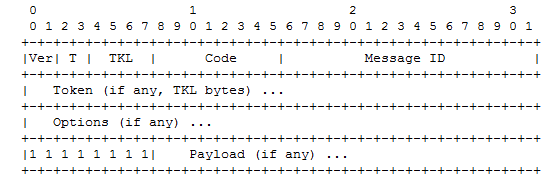


شکل 9 ساختار پروتکلCoAP

از آنجایی که پروتکل UDP امکان انتقال قابل‌اطمینان پیغام‌ها را فراهم نمی‌سازد، این وظیفه در بخش Message با استفاده از الگوریتم عقب‌گرد نمایی[[13]](#footnote-13) پیاده‌سازی می‌شود. قسمت Request/Response نیز درخواست‌های REST را مدیریت می‌کند. این درخواست‌ها به وسیله پیغام‌های CoAP ارسال می‌شوند و شامل URI منبع هستند.

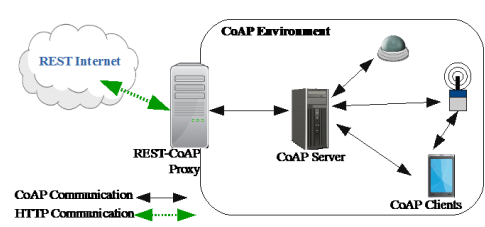
در CoAP چهار نوع پیغام تعریف شده است: confirmable، non-confirmable، reset و acknowledgment. قابلیت اطمینان توسط پیغام‌های confirmable پیاده‌سازی می‌شود که دارای یک زمان time-out هستند و باید توسط پیغام acknowledgement رسیدن آن تایید شود. بنابراین در صورتی که پیغام confirmable باشد،گیرنده یک پیغام acknowledgement ارسال می‌کند که نشان‌دهنده تایید دریافت پیغام است. پیغام‌ها می‌توانند به صورت non-confirmable نیز ارسال شوند که دیگر برای آن‌ها پیغام acknowledgementارسال نمی‌شود. در صورتی که گیرنده نتواند پیغام دریافتی را مورد پردازش قرار دهد یک پیغام reset ارسال می‌کند. پیغام acknowledgement را می‌توان به صورت جدا و یا همراه با پیغام response ارسال کرد.

ساختار کلی یک پیغام در **Error! Reference source not found.Error! Reference source not found.** نشان داده شده است. بخش Ver بیانگر نسخه پروتکل مورد استفاده است. بخش T بیانگر نوع پیغام است که یکی از چهار نوع گفته‌شده می‌تواند باشد. بخش TKL بیانگر طول Token است. از Token در Request/Response به منظور تطبیق دادن Request به Response استفاده می‌شود و مستقل از Message ID است. Message ID به منظور ایجاد قابلیت اطمینان در لایه Messages اضافه می‌شود. کدهای مربوط به Request مانند GET،PUT، POST و DELETE و یا کدهای Response مانند Bad Request و Not Found در بخش Code قرار می‌گیرند.



شکل 10 فرمت کلی پیغام CoAP

در نهایت می‌توان محرمانگی و یکپارچگی پیغام‌های ارسالی را با استفاده از پروتکل DLTS که در لایه بالای پروتکل UDP قرار می‌گیرد تضمین کرد. یک سناریو از پیاده‌سازی این پروتکل در محیطIoT در **Error! Reference source not found.** مشاهده می‌شود.

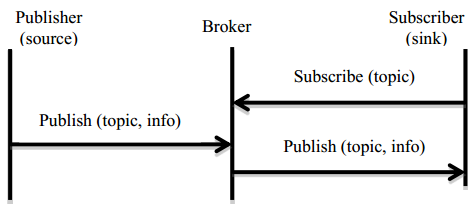


شکل 11 سناریوی پیاده‌سازی CoAP

### پروتکل MQTT

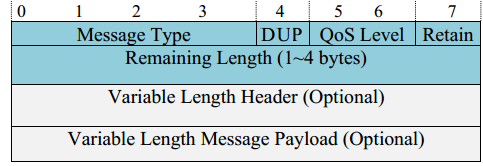
پروتکل MQTT در سال 1999 توسط IBM پیشنهاد شد و در سال 2014 نسخه 3.1.1 آن توسط OASIS استاندارد شده است. اگرچه در کاربردهای IoT نسخه 1.2 آن که مناسب برای شبکه‌های حسگر بی‌سیم است بیشتر مورد توجه قرارگرفته است. این پروتکل مبتنی بر معماری Publish-Subscriber بوده و مناسب برای گره‌هایی با منابع محدود و شبکه‌هایی با لینک‌های ناپایدار با پهنای باند کم است. این پروتکل از سه سطح کیفیت سرویس حمایت می‌کند و پروتکل لایه حمل آن نیز TCP است.

پروتکل MQTT از سه جز تشکیل شده است: Publisher، Subscriber و Broker. گره‌ها می‌توانند به عنوان Subscriber در Broker خود را برای موضوعات خاصی ثبت‌نام کنند تا هنگامی که مطالب مرتبط به آن‌ها از سمت Publisher ها منتشر شد، از طریق Broker به آن‌ها اطلاع داده شود. در نهایت اطلاعات از طریق Broker به سمت Subscriber ارسال می‌شود. البته Broker می‌تواند بحث‌های امنیتی مانند احراز هویت و یا سطح دسترسی اطلاعات را نیز کنترل نماید. معماری ساده شده این پروتکل در **Error! Reference source not found.** نمایش داده شده است.



شکل 12 معماری MQTT

در نسخه 1.2 این پروتکل بیش از 20 نوع پیغام تعریف شده است که ساختار کلی این پیغام‌ها در **Error! Reference source not found.** نمایش داده شده است.



شکل 13 ساختار کلی پیغام MQTT

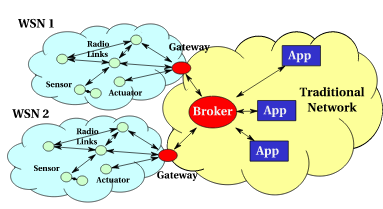
Message Type بیانگر نوع پیغام است که می‌تواند Connect،Publish ، Subscribe و ... باشد. DUP بیانگر تکراری بودن پیغام است. QoS Level سطح کیفیت سرویس را مشخص می‌کند. سه سطح کیفیت سرویس پشتیبانی شده عبارتند از:

سطح صفر: پیغام‌ها ممکن است به مقصد تحویل داده شوند و یا گم شوند و تلاشی برای تکرار نخواهد شد.

سطح یک: پیغام‌ها به مقصد تحویل داده خواهند شد ولی ممکن است پیغام تکراری نیز تحویل داده شود

سطح دو: دقیقا یک کپی از پیغام حتما به مقصد تحویل داده خواهد شد.

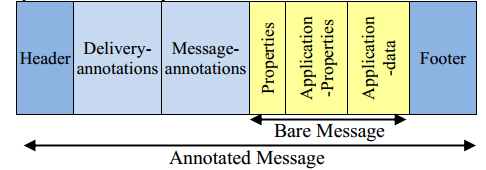
بخش Retain به Broker می‌گوید که آخرین پیغامPublish را ذخیره کرده و به‌عنوان اولین پیغام به گره‌های جدیدی که Subscribe می‌کنند ارسال کند. یک سناریو از پیاده‌سازی MQTT در محیط حسگر بی‌سیم در **Error! Reference source not found.** نمایش داده شده است.



شکل 14 سناریوی پیاده‌سازی MQTT

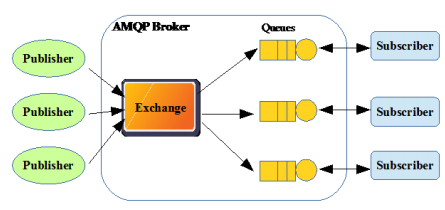
### پروتکل AMQP

پروتکل AMQP نسخه 1 مبتنی بر معماری Publish-Subscriber است و نیازمند یک پروتکل لایه حمل قابل‌اطمینان مانند SCTP یا TCP است. بخش Broker در این پروتکل از دو جز Exchange وMessage Queue تشکیل شده است. Exchange برای مسیریابی پیغام‌ها به سمت صف‌های مرتبط مورد استفاده قرار می‌گیرد. این مسیریابی می‌تواند بر اساس قواعد از قبل تعریف شده باشد. سپس از صف‌ها، پیغام‌ها به سمت دریافت‌کنندگان پیغام ارسال می‌شوند. ارسال می‌تواند به صورت point-to-point به سمت یک مقصد خاص ارسال شوند و یا اینکه مطابق سناریوی Publish-Subscribe پیغام به سمت Subscriber ها ارسال شوند. در AMQP دو نوع پیغام وجود دارد، Bare و Annotated که ساختار این دو پیغام در **Error! Reference source not found.** مشاهده می‌شود.



شکل 15 ساختار پیغام در AMQP

همان‌گونه که در **Error! Reference source not found.** مشاهده می‌شود، پیغامAnnotated که در گیرنده دریافت می‌شود، در حقیقت همان پیغام Bare است که توسط فرستنده ارسال می‌شود که سرآیند و Footer به آن اضافه ‌شده است. Header بخش‌هایی مانند TTL، اولویت تحویل پیغام و Durability را مشخص می‌کند. اگر پیغامی Durable باشد نباید Lost شود. با استفاده ازDelivery Annotations می‌توان اطلاعات غیراستاندارد از پیش تعریف شده بین فرستنده و گیرنده را مبادله کرد. Message Annotation نیز شامل اطلاعات غیراستانداردی است که هر گیرنده‌ای در مسیر می‌تواند آن را استفاده کند. Properties شامل اطلاعاتی مانند ID پیغام، فرستنده و گیرنده پیغام و موضوع پیغام است. Application Properties شامل اطلاعات کاربرد است که ساختار مشخصی دارند و Application Data شامل اطلاعات باینری است. همچنین Footer می‌تواند شامل اطلاعاتی مانند امضای دیجیتال و یا هش پیغام باشد{Citation}. یک نمونه از سناریو Publish-Subscriberدر **Error! Reference source not found.** مشاهده می‌شود.

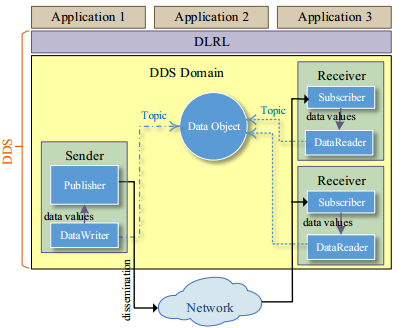


شکل 16 ارسال پیغام به صورتPublish-Subscribe

### پروتکل DDS

پروتکل DDS نیز مانند AMQP و MQTT یک پروتکل مبتنی بر معماری Publish-Subscribe است با این تفاوت که در این پروتکل Broker وجود ندارد و از چندپخشی برای تحویل پیغام‌ها و پشتیبانی از کیفیت سرویس استفاده می‌کند. این پروتکل از 23 نوع سیاست کیفیت سرویس مطابق با شرایط مختلف پشتیبانی می‌کند.

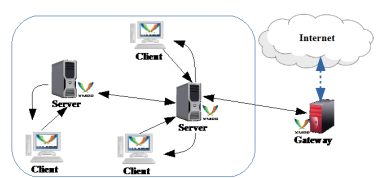
این پروتکل از دو بخش تشکیل شده است: DCPS[[14]](#footnote-14) و DLRL[[15]](#footnote-15). بخش DCPS مسئول تحویل پیغام‌ها به Subscriber ها را بر عهده دارد و بخش DLRL مانند یک واسط برای دسترسی به توابع DCPS مورد استفاده قرار می‌گیرد. در حقیقت نقش آسان‌سازی به اشتراک‌گذاری داده‌های توزیع‌شده در بین اشیا توزیع‌شده را بر عهده دارد. در بخش DCPS بخش‌های فرستنده، گیرنده و Topic وجود دارد. بخش فرستنده از Publisher که داده‌ها را منتشر می‌کند و DataWriter تشکیل شده است. DataWriter توسط کاربرد برای ارتباط با Publisher مورد استفاده قرار می‌گیرد و داده‌ها و نوع آن‌ها را برای Publisher مشخص می‌کند. بخش گیرنده از Subscriber و DataReader تشکیل شده است که DataReader داده‌های دریافت شده توسط Subscriber را تحلیل می‌کند. بخش Topic دو بخش DataWiter و DataReader را به یکدیگر مرتبط می‌کند و هر Topic به ‌وسیله اسم و نوع داده مشخص می‌شود. ارسال و دریافت اطلاعات، داخل یک دامنه DDS که یک محیط مجازی برای کاربردهای در ارتباط با یکدیگر است محقق می‌شود[18]. نمونه‌ای از معماری مفهومی DDS در شکل زیر نمایش داده شده است.



شکل 17- معماری مفهومیDDS

### پروتکل XMPP

پروتکل XMPP توسط IETF به منظور پیغام‌رسانی و[[16]](#footnote-16)IM در اینترنت استاندارد شده است. این پروتکل از احراز هویت و رمزنگاری انتها به انتها نیز پشتیبانی می‌کند. این پروتکل مبتنی بر معماری مشتری-سرویس دهنده است. در این پروتکل پیغام‌ها از رشته‌های XML تشکیل شده‌اند که شامل سه بخش Message، Presence و Info/query است. بخش Message شامل فرستنده، گیرنده و محتوای پیغام است. با استفاده از بخش Presence، وجود داشتن و در دسترس بودن یک موجودیت خاص اعلام می‌شود. همچنین می‌توان انواع وضعیت‌های مختلف مانند Busy بودن فرستنده و یا گیرنده را نیز اعلام کرد. پیغام‌های Request-Response در بخش Info/query قرار می‌گیرند. به عنوان مثال زمانی که لازم باشد نشست جدید ایجاد شود، درخواست ایجاد نشست و جواب آن در این قسمت قرار می‌گیرد[18]. نمونه‌ای از یک سناریوی XMPP در شکل زیر مشاهده می‌شود.



شکل 18- سناریوی XMPP[18]

### جمع‌بندی

تحقیقات متعددی در زمینه کارایی پروتکل‌های بررسی‌شده صورت گرفته است. درکارایی دو پروتکل CoAP و MQTT ازنظر پهنای باند مورد استفاده و تاخیر انتها به انتها مورد بررسی قرار گرفته است و نشان داده شده است که در زمانی که نرخ از دست رفتن بسته در شبکه پایین باشد، پروتکل MQTT کارایی بهتری را از خود نشان می‌دهد. این در حالی است که در نرخ‌های حذف بسته بالا، کارایی CoAP از MQTT بهتر است که این نتایج مطابق با بررسی صورت گرفته در است. درکارایی AMQP با REST از نظر تعداد پیغام‌های رد و بدل شده بررسی شده و نشان داده شده است که AMQP کارایی بهتری از خود نشان می‌دهد. در جدول زیر ویژگی‌های پروتکل‌های لایه کاربرد بررسی‌شده با یکدیگر مقایسه شده است.

جدول 1 پروتکل‌های لایه کاربرد

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| پروتکل | پشتیبانی از REST | پروتکل لایه حمل پیشنهاد شده | معماری | امنیت |
| CoAP | دارد | UDP | مشتری- سرویس‌دهنده | DTLS |
| MQTT | ندارد | TCP | Publish-Subscriber | TLS |
| AMQP | ندارد | TCP | Publish-Subscriber | TLS |
| DDS | ندارد | TCP-UDP | Publish-Subscriber | TLS-DTLS |
| XMPP | ندارد | TCP | مشتری- سرویس‌دهنده | TLS |

# فصل چهارم

## بیگ دیتا چیست؟

در تحقیقات اولیه، در سایت های گوناگون، عموماً به این تعریف کلی برمی‌خوردیم که به داده های با حجم بیش از چندین هزار ترابایت داده حجیم گویند. اما این تعریف ناقص است.

بطور کلی داده های حجیم از سه منظر بررسی می‌شوند، در واقع با داشتن هر کدام از این سه ویژگی داده حجیم محسوب می‌شود.

سرعت :سرعت یک داده از منظر تولید داده و یا سرعت انتقال(حرکت) آن. مانند سرعت تولید و حرکت پیام های شبکه های اجتماعی.

تنوع :اشاره به انواع مختلف داده دارد که ما امروزه از آن‌ها استفاده می کنیم. در گذشته تنها با داده هایی ساخت یافته سر و کار داشتیم که قابل نمایش و ذخیره سازی در پایگاه داده و جداول بودند، همانند داده‌های مالی. اما امروزه بیش از ۸۰ درصد از داده‌ها را، داده‌های غیر ساخت یافته، مانند عکس و فیلم و … تشکیل می‌دهند.

حجم: اشاره به مقادیر زیادی که در هر ثانیه تولید می‌شود. این مقادیر زیاد داده در حدود زتابایت و برونتوبایت می‌باشد.

اریک اشمیت، مدیر اجرایی گوگل می‌گوید:

“از سپیده دم تمدن تا سال 2003، بشر پنج اگزابایت داده تولید کرده است. در حال حاضر ما در هر دو روز پنج اگزابایت تولید می‌کنیم و این سرعت تولید داده شتاب گرفته است”.

## بیگ دیتا و اینترنت اشیا

اما سؤال اینجاست که داده‌های اینترنت اشیا از چه نوعی بیگ دیتا محسوب می‌شوند؟ داده‌های بیگ دیتا عموماً ۲ ویژگی از ویژگی های نام برده را توأمان دارند. که این ۲ ویژگی اکثراً سرعت و حجم بالا می‌باشد.

اما چطور این داده‌های حجیم

در واقع اگر هر کدام از این وجوه را به تنهایی برای تعریف داده های حجیم بکار بریم مانند انسانی خواهیم بود که با چشمان بسته قصد تشخیص یک فیل را دارد.

اما سؤال اینجاست که بیگ دیتا چه نقشی در اینترنت اشیا دارد؟

• اینترنت اشیا در واقع به چیزهای مرتبط به هم گویند، و ما با مانیتور کردن پیکربندی، رفتار این اشیا، حالاتشان، مشکلاتشان و غیره

• تعمیر مشکلات پیش بینی شده :‌ پیش از خراب شدن، مشکل را پیدا کرده و درصدد حل آن برآییم

• بهینه سازی: پیکربندی، تعامل با انسان ها، بهره وری انرژی، مناسب ترین "چیز" برای استفاده های مختلف

• تجزیه و تحلیل برای پیدا کردن کاستی فعلی برای طراحی نسخه های بعدی از "چیز‌"ها

از قابلیت های مهم اینترنت اشیا، بدست آوردن بینش‌های مهم، بهینه سازی فرآیند‌ها و غیره با مشاهده رفتار “چیز” ها است. این قابلیت به حل چالش‌های متفاوتی ریز می‌شود:

ذخیره‌سازی تمامی حوادث و حالات(چالشی برای سرعت و حجم)

اجرا کردن جست و جوهای آنالیزی بر روی وقایع ذخیره شده(چالشی برای سرعت و حجم)

تجزیه و تحلیل داده‌ها به منظور کسب بینش درباره داده‌ها(چالشی برای سرعت و حجم و تنوع داده)

از دیگر قابلیت های مهم اینترنت اشیا، توانایی انجام آنالیز برخط است.

چگونگی تشخیص و برخورد با فرصت ها و تهدید‌ها کسب و کار

که این قابلیت نیز نیازمندی‌های متفاوتی دارد:

چگونگی پردازش جریان رویداد‌های برخط(چالشی برای سرعت)

چگونگی ذخیره سازی

## نحوه برخورد مناسب با بیگ دیتا

پس از آشنایی با داده‌های حجیم و انواع آن و علی الخصوص داده‌های حجیم در حوزه اینترنت اشیا، به دنبال راهکاری برای تجزیه و تحلیل آن بودیم. پس از تحقیق بسیار به انتخاب یکی از دو فریم ورک هدوپ و اسپارک رسیدیم. به دلیل اینکه هدوپ بیشتر در این زمینه سابقه داشته و کار‌های انجام شده و متن باز زیادی داشت، ابتدا آن را برای ادامه کار خود انتخاب کردیم.

## هدوپ

هادوپ یک منبع باز، چارچوب برنامه نویسی مبتنی بر جاوا است که پردازش و ذخیره سازی از مجموعه داده های بسیار بزرگ در یک محیط محاسبات توزیع شده پشتیبانی می کند. این بخشی از پروژه آپاچی با حمایت مالی بنیاد نرم افزار آپاچی است.

در حالی که شبکه جهانی وب در اواخر 190۰ و 2000 اولیه رشد کرد، موتورهای جستجو و شاخص ها برای کمک به مشخص کردن اطلاعات مربوطه در میان محتوای مبتنی بر متن ایجاد شد. در سال های اولیه، نتایج جستجو توسط انسان بازگردانده می‌شدند. اما با رشد وب از ده ها به میلیون ها صفحه، اتوماسیون مورد نیاز بود. خزنده وب، ایجاد شد که بسیاری به عنوان پروژه های تحقیقاتی دانشگاهی در پی آن بودند، و استارت آپ های موتور جستجو شکل گرفتند (یاهو، آلتاویستا، و غیره).

یکی از پروژه های شکل گرفته، پروژه ناچ بود که زاییده ذهن Doug Cutting و Mike Cafarella بود. آنها می خواستند نتایج جستجوی وب را با توزیع داده ها و محاسبات در رایانه‌های مختلف سریعتر برگردانند تا وظایف متعدد بتوانند به طور همزمان انجام شوند. در همین زمان، یکی دیگر از پروژه‌های موتور جستجو به نام گوگل در حال پیشرفت بود. آن را بر روی همان مفهوم - ذخیره سازی و پردازش داده‌ها بطور توزیع شده، و خودکار به طوری که نتایج جستجو‌های مرتبط در وب بتوانند سریع تر برگردند.

در سال 2006، آقای کاتینگ به یاهو پیوست و پروژه ناچ و همچنین ایده های مبتنی بر کارهای اولیه گوگل را با خود به همراه ذخیره سازی توزیع شده خودکار و پردازش داده ها به یاهو برد. این پروژه ناچ تقسیم شد - قسمت خزنده وب به عنوان ناچ باقی ماند و بخش محاسبات و پردازش توزیع شده هدوپ شد (به نام فیل اسباب بازی پسر کاتینگ است). در سال 2008، یاهو هادوپ به عنوان یک پروژه منبع باز منتشر شد. امروز، چارچوب و اکوسیستم از فن آوری های هادوپ توسط بنیاد نرم افزار آپاچی (ASF)، یک جامعه جهانی از توسعه دهندگان نرم افزار و همکاران مدیریت و نگهداری می‌شود.

### اهمیت هدوپ

• قابلیت ذخیره و پردازش حجم زیادی از هر نوع داده، با سرعت بالا. با حجم و انواع روز افزودن داده‌ها، به ویژه از رسانه های اجتماعی و اینترنت اشیاء، که در نظر گرفتن کلید است.

• قدرت پردازش. مدل محاسبات توزیع شده هدوپ پردازش داده های بزرگ را سریع می کند. هرچقدر گره های محاسباتی بیشتری استفاده کنید، قدرت پردازش بیشتری خواهید داشت.

• تحمل خطا. داده ها و پردازش نرم افزار در برابر شکست سخت افزار محافظت می شود. اگر یک گره پایین رود، شغل به طور خودکار به گره های دیگر هدایت می شود تا مطمئن شود که محاسبات توزیع شده شکست نمی‌پذیرد. نسخه های متعدد از تمام داده ها به طور خودکار ذخیره می شود.

• انعطاف پذیری. بر خلاف پایگاه داده های رابطه ای سنتی، شما مجبور به پردازش اطلاعات قبل از ذخیره سازی ندارید. شما می توانید داده ها آنجا که می خواهید ذخیره کنید و بعداً تصمیم بگیرید که چگونه از آن استفاده کنید . این داده‌ها شامل داده های بدون ساختار مانند متن ها، تصاویر و فیلم ها هم می‌شود.

• کم هزینه. فریم ورک منبع باز رایگان است و از سخت افزار مناسب برای ذخیره مقادیر زیادی از داده ها استفاده می‌کند.

• مقیاس پذیری. شما به راحتی می توانید با اضافه کردن گره، داده‌های بیشتری را در سیستم خود ذخیره کنید.

### چالش های استفاده از هادوپ

برنامه نویسی نگاشتکاهش یک روش خوب برای تمام مشکلات نیست. این روش برای درخواست اطلاعات ساده و مشکلاتی که می توان آن‌ها را به واحدهای مستقل تقسیم کرد، مناسب است، اما برای انجام دادن کارهای تحلیلی تکرار شونده و تعاملی کارآمد مناسب نیست.

یک فاصله استعدادی گسترده ای در کار با نگاشتکاهش وجود دارد. یافتن برنامه نویسان تازه‌کاری که مهارت‌های کافی جاوا را دارند برای کار با نگاشتکاهش مشکل است. این یکی از دلایلی است که ارائه دهندگان توزیع در حال تلاش برای قرار دادن رابط تکنولوژی (SQL) در بالای هادوپ هستند. یافتن برنامه نویسان با مهارت های SQL بسیار آسان تر از برنامه نویسان با مهارت های نگاشتکاهش است.

امنیت داده ها. چالش دیگر حول مشکل امنیت قسمتی داده‌ها ست، البته ابزار‌های جدیدی به این عرصه وارد شده‌اند. پروتکل احراز هویت Kerberos یک گام بزرگ به سوی ساخت محیطی امن برای هدوپ است.

مدیریت , حاکمیت کامل داده ها. هادوپ ابزارهایی با ویژگی های کامل و آسان برای استفاده برای مدیریت داده ها، پاک کردن داده، حاکمیت و ابرداده ندارد.

### اینترنت اشیا و هدوپ

همه چیز در اینترنت اشیا باید بدانند با چه چیز و در چه زمانی در ارتباط باید باشند. در هسته اینترنت اشیا جریانی، همیشه در تورنت داده ها در جریان است. هدوپ اغلب به عنوان ذخیره کننده داده ها برای میلیون ها و یا میلیاردها داد و ستد داده‌ای استفاده می شود. قابلیت های عظیم ذخیره سازی و پردازش نیز اجازه می دهد از هدوپ به عنوان یک سندباکس برای کشف و تعریف الگوهایی برای مشاهده آموزش تجویزی استفاده شود. بعد از آن شما می توانید به طور مداوم این دستورالعمل را بهبود بدهیم، زیرا هدوپ به طور مداوم با داده های جدید که با الگوریتم‌های قبلا تعریف شده مطابقت ندارند به روز می‌شود.

## اسپارک

آپاچی اسپارک موتور پردازش منبع باز قدرتمند ساخته شده حول سرعت، سهولت استفاده، و تجزیه و تحلیل پیچیده است. این موتور پردازش در اصل در دانشگاه برکلی در سال 2009 توسعه داده شد.

بزرگترین پروژه منبع باز در پردازش داده ها.

از زمان انتشار خود، آپاچی اسپارک تصویب سریعی را توسط شرکت‌ها در سراسر طیف گسترده ای از صنایع را دیده است. موتورخانه اینترنت مانند Netflix، یاهو، و ای بی از اسپارک در مقیاس گسترده بهره برده‌اند، مجموعاً پردازش پتابایت‌های چندگانه‌ای از داده ها در خوشه هایی بیش از 8000 گره. اسپارک به سرعت به بزرگترین جامعه منبع باز در داده های بزرگ، با بیش از ۱۰۰۰ همکاران از بیش ۲۵۰ سازمان تبدیل شد.

موتور پردازش؛ به جای فقط "نقشه" و "کاهش"، تعریف یک مجموعه بزرگ از عملیات (تحولات و اقدامات)

مجموعه داده توزیع شده انعطاف پذیر(RDD) - ساختار کلیدی اسپارک

RDDنماینده داده ویا دگرگونی بر روی آن است

RDDمی‌تواند از ورودی های به فرمت هدوپ و یا دگرگونی سایر RDD ها ساخته شود.

روی این RDD ها می‌تواند اعمالی صورت گیرد، بدین ترتیب که اعمال ریاضی بر رویشان انجام شده و مقدار برمیگردانند.

ارزیابی تنبل :‌تا زمانی که به نتیجه یک محاسبات نیازی نباشد، آن را انجام نمی‌دهد.

RDDبرای نرم افزار‌هایی که یک عمل یکسان بر المان‌های مختلف آن انجام می‌شود مناسب است.

دلیل انتخاب اسپارک بر هدوپ

ما ابتدا هدوپ را برای این پروژه انتخاب کردیم، اما پس از مطالعه و پیشرفت در پروژه، به چند مسئله برخوردیم. مسئله اول، نحوه ارتباط بستر تینگ تاک به قسمت آنالیزور سرور بود، که با هدوپ ایجاد این ارتباط بسیار دشوار بود. مسئله دوم برتری سرعتی اسپارک بر هدوپ بود. با تحقیقات بیشتر بر روی اسپارک، به چندی از تفاوت‌هایشان اشاره می‌کنیم که باعث تغییر فریم ورک پروژه ما نیز شد.

عملکرد : اسپارک عموماً سریعتر است.

اسپارک می تواند داده‌ها را در حافظه نیز پردازش کند، اما هدوپ بلافاصله داده‌ها را به دیسکبرمیگرداند.

راحتی استفاده :‌ اسپارک برای برنامه نویسی راحت تر است.

اسپارک نسخه کامل شده‌ای از نگاشتـکاهش هدوپ است، پس کامل تر از آن است.

### مزیت‌های اسپارک

• سرعت

مهندسی از پایین به بالا برای عملکرد، اسپارک می تواند ۱۰۰ مرتبه سریع تر از هدوپ برای پردازش داده ها در مقیاس بزرگ با استفاده از محاسبات درون حافظه و بهینه سازی های دیگر عمل کند. اسپارک نیز هنگامی که داده ها بر روی دیسک ذخیره شده‌اند سریع است، و در حال حاضر دارای رکورد جهانی برای مرتب سازی مقیاس بزرگ بر روی دیسک است.

• راحتی در استفاده

اسپارک رابط های کاربری آسان برای استفاده‌ای برای کار در مجموعه داده های بزرگ دارد. این شامل مجموعه ای از بیش از 100 اپراتور برای تبدیل داده ها و رابط های کاربری قاب داده‌ای آشنا برای دستکاری داده های نیمه ساخت دارد.

• یک موتور یکپارچه

اسپارک همراه با کتابخانه های سطح بالا، از جمله پشتیبانی از جست و جوهای SQL، جریان داده ها، یادگیری ماشین و پردازش نمودار بسته بندی است. این کتابخانه‌های استاندارد افزایش بهره وری توسعه دهنده را به همراه دارد و می تواند برای ایجاد گردش کار پیچیده ترکیب شود.

### اکوسیستم آپاچی اسپارک

• داده ساخت یافته: Spark SQL

بسیاری از کاربران برای جست و جو در داده‌ها، از جست و جو های sql بهره می‌برند. اسپارک اسکیوال ماژولی ست برای تحلیل داده‌های ساخت یافته.

• تحلیل جریان داده‌ای: Spark Streaming

بسیاری از برنامه‌ها، مانند مسائل و برنامه‌های دخیل در اینترنت اشیا، نیاز دارند تا تحلیل‌های لازم بر روی جریان داده‌ای انجام گیرد. اجرا شده بر روی اسپارک، این اجازه را به ما می دهد تا بتوانیم داده‌های خود را هم بصورت ذخیره شده در حافظه و هم بصورت جریانی تحلیل کنیم. آن را به آسانی با طیف گسترده ای از منابع داده‌ای محبوب، از جمله HDFS، فلوم، کافکا و توییتر می‌توان ادغام کرد.

یادگیری ماشین :MLlib

یادگیری ماشین به عنوان جزئی اصلی در ساختار و تحلیل بیگ دیتا آمده است. این کتابخانه یادگیری ماشین بر روی اسپارک ساخته شده است و هم الگوریتم‌های متنوعی را پوشش می‌دهد و هم از سرعت بسیار بالایی برخوردار است. این کتابخانه زبان‌های جاوا، اسکالا و پایتون را هم پوشش می‌دهد.

محاسبات گراف : GraphX

گراف ایکس یک موتور محاسبه گراف است که برو روی اسپارک ساخته شده است و کتابخانه‌ای کامل از الگوریتم‌ها را به همراه دارد. این موتور به کاربر اجازه می‌دهد تا داده‌های با ساختار گراف را بسازند، تحلیل و منتقل کنند.

اجرای عمومی: هسته اسپارک

هسته اسپارک زمینه موتور اجرایی عمومی برای پلت فرم اسپارک است که تمام قابلیت های دیگر در بالای آن ساخته شده است. این هسته در حافظه قابلیت محاسبات برای ارائه سرعت، مدل اجرای عمومی برای پوشش از یک طیف گسترده ای از برنامه های کاربردی، و API های جاوا، اسکالا، و پایتون برای سهولت توسعه را فراهم می کند.

## درباره فلوم

فلوم خدمتی توزیع شده، قابل اعتماد و در دسترس برای جمع آوری موثر، ادغام، و انتقال مقادیر زیادی از داده‌های لاگ است. فلوم معماری ساده و انعطاف پذیری بر اساس جریان داده ها استریم دارد.. این سرویس قوی و تحمل پذیر خطا(fault tolerant) با مکانیسم های قابلیت اطمینان موزون و بسیاری از مکانیزم بهبود است. با استفاده از یک مدل داده های درب ساده است که استفاده های تحلیلی آنلاین را ممکن می‌سازد.

## پروژه ما

با توجه به پروژه گروه های دیگرکه شهر هوشمند بود، و گروه‌ها داده‌های خود را بر روی بستر thingSpeak میفرستاد که برای آنالیز داده‌ها نیاز بود با استفاده از فلوم آن را به اسپارک متصل کرده و از طریق آن داده ها را پردازش کنیم. که از طریق یک کتابخانه زبان اسکالا در فریم ورک اسپارک، این اتصال انجام شد. در ادامه داده‌ها را ابتدا در حافظه بطور hdfs ذخیره کردیم. این عمل ذخیره همواره باید انجام پذیرد، همانطور که سنسور‌ها همواره در خال ارسال داده برای ما هستند، ما آن‌ها را ذخیره می‌کنیم. سپس آنالیز را انجام داده و نتیجه نهایی را بر روی thingspeak فرستادیم. لازم به ذکر است کد پروژه به زبان اسکالا زده شده است. دلیل انتخاب این زبان functional بودن آن بود و همنیطور زبان پیشنهادی خود اسپارک است.

عملیات عمومی تا بحال انجام شده، میانگین گیری از داده هاست.

روند کلی هم به همین ترتیب نوشته شده در گزارش بود. بدین ترتیب که ما ابتدا تحقیقات خود را بر روی اینترنت اشیا انجام دادیم. سپس با مطالعه بر روی بزرگ داده و چکونگی تحلیل آن ادامه دادیم. در این مرحله ابتدا هدوپ را انتخاب کردیم و مراحل نصب و آماده‌سازی و اتصال آن را به بستر داده‌ها طی کردیم، اما با توجه به سختی‌های کار و احساس کمبود در آن، به اسپارک روی آوردیم. پس از انتخاب اسپارک، برای اتصال آن به بستر و دریافت داه‌ها مجبور به استفاده از فلوم شدیم، و در نهایت کد خود را در اسپارک به زبان اسکالا پیاده سازی کردیم.

## کارهای آینده

تا اینجا ما مراحل اولیه برای آنالیز داده های را انجام داده ایم، اما برای آنالیز واقعی، نیازمند داده هایی واقعی هستیم تا بتوانیم نتایجی واقعی را نشان دهیم. این مرحله نیازمند تکمیل پروژه های دیگر است که هنوز انجام نشده است. اما پروژه ما آماده بهره برداری بوده و داده ها را ابتدا بصورت json ذخیره کرده و سپس می‌توان محاسبات ریاضی مورد نیاز را انجام داد و نتایج خروجی را به شکل مورد نیاز نشان دهد.

برای نشان دادن آنچه مدعی هستیم از نمونه کارهای انجام شده مثال می‌آوریم.

## پروژه های انجام شده

تا به امروز پروژه‌های زیادی مانند پروژه ما انجام گرفته است، اما تنها تعداد معدودی به اجرا درآمده اند و در واقعیت با داده‌های واقعی آزمایش شده‌اند. در اینجا به معرفی و بررسی دو پروژه از این دست می‌پردازیم.

بطور کلی پروژه‌های انجام گرفته عموماً در حوزه صنعت، شهر هوشمند، انرژی هوشمند و ماشین‌های مرتبط می‌باشد. حدود ۲۰ درصد از پروژه‌ها در حوزه شهر هوشمند می‌باشند و تا به امروز در شهر‌های متعددی در جهان و به خصوص در اروپا به اجرا درآمده‌اند. ۱۰ شهر اول در این حوزه عبارتند از:

• کپنهاگن

• استکهلم

• آمستردام

• وین

• پاریس

• برلین

• لندن

• بارسلونا

• مونیخ

• فرانکفورت

ما برای نمونه به بررسی شهر هوشمند بارسلونا می‌پردازیم.

پروژه‌های انجام شده در این شهر بدین قرار است:

• بستر های نرم افزاری شهری(City OS،‌ بستر نرم افزاری سنسور‌های بارسلونا)

• . اطلاعات هوشمند (باز کردن داده ها، اتاق وضعیت)

• وای فای شهری

• مدیریت آبیاری از طریق تلفن همراه

• طرح مدیریت روشنایی

• جزایر خود کفا

• توسعه وسیله نقلیه الکتریکی

• شبکه متعامد اتوبوس

• جیب بارسلونا (بارسلونا بدون تماس و برنامه های تلفن همراه)

• پروژه های اروپایی

که بطور مثال اگر بخواهیم به بررسی طرح مدیریت سبکه اتوبوس بپردازیم، پروژه ای مانند پروژه پیشنهادی یکی از گرو‌ه‌های ما می‌باشد. که بر رفت و آمد اتوبوس‌ها نظارت می‌کند و از وضعیت مسافران آن، میزان شلوغی خطوط، مکان هر اتوبوس و ... ما را مطلع می‌کند.

و در طرح مدیریت روشنایی، در هر منطقه لامپ به میزان کافی روشن شده و نور آن به میزان لازم تنظیم می‌شود.

از پروژه‌های دیگری که روی آن‌ها سازمان‌های مختلفی کار کرده‌اند، پروژه‌های healthcare را می‌توان نام برد. که این نوع پروژه را ما نیز در پروژه‌های خود داشتیم.

از جمله این دسته از پروژه‌های اینترنت اشیا می‌توان به پروژه برند پولو و مقاله ای با طرح اجرایی آزمایش شده پرداخت.

در پروژه برند پولو، این برند با تولید لباسی ورزشی، سنسوری در آن قرار داده است که وضعیت جسمی و فعالیت‌های او را در طول روز ذخیره کرده و آن را در گوشی تلفن همراه فرد به نمایش می‌گذارد.

در مقاله‌ای هم طرحی مورد آزمایش قرار گرفته‌ است، که وضعیت جسمی و فعالیت‌ها فرد را طول روز ذخیره کرده و می‌خواهد بطور میانگین میزان فعالیت افراد را در طول روز ببیند. با اجرای طرح بر روی ۳۰ میلیون نفر، نتایج زیر بدست آمده است.

پیوست‌ها

1. Kelvin Ashton [↑](#footnote-ref-1)
2. International Telecommunication Unit [↑](#footnote-ref-2)
3. User Context [↑](#footnote-ref-3)
4. Peer to Peer [↑](#footnote-ref-4)
5. IoT Application Enablement Platform [↑](#footnote-ref-5)
6. IaaS backends [↑](#footnote-ref-6)
7. Hardware-specific software platform [↑](#footnote-ref-7)
8. Enterprise software extensions [↑](#footnote-ref-8)
9. Event-action-trigger [↑](#footnote-ref-9)
10. Constrained RESTful Environments [↑](#footnote-ref-10)
11. Unified Resource Identifier [↑](#footnote-ref-11)
12. Multicast [↑](#footnote-ref-12)
13. Exponential Backoff [↑](#footnote-ref-13)
14. Data-Centric Publish-Subscribe [↑](#footnote-ref-14)
15. Data-Local Reconstruction Layer [↑](#footnote-ref-15)
16. Instant Messaging [↑](#footnote-ref-16)