پروژه درس معماری پیشرفته

مروری بر تحقیقات گذشته

استاد:

خانم دکتر جاسبی

دانشجو:

ندا باغبان کاشانی

(۳۹۹۱۲۳۴۱۰۵۷۰۲۴)

خردادماه ۱۴۰۰

عنوان مقاله:

An architecture for adaptive task planning in support of IoT-based machine learning applications for disaster scenarios

یک معماری برای برنامه‌ریزی وظیفه تطبیقی در پشتیبانی از برنامه‌های کاربردی یادگیری ماشین مبتنی بر IoT برای سناریوهای فاجعه

(Sacco et al., 2020) – Computer Communications

تعریف مساله و هدف اصلی مقاله:

در سال‌های اخیر شاهد گسترش محاسبات موبایل و اینترنت اشیا (IoT) بوده‌ایم، که در آن میلیاردها دستگاه تلفن همراه و IoT به اینترنت متصل شده‌اند، و مجموعه داده‌های بزرگی را ایجاد می‌کنند که باید توسط چندین برنامه (توزیع‌شده) مصرف شوند. یک زیرمجموعه از این کاربردها نیاز به این دارد که دستگاه‌های IoT به طور جداگانه برنامه‌ریزی شوند تا یک ماموریت را به طور مستقل انجام دهند. نمونه‌های معمول چنین سناریوهایی، شبکه‌های ناهمگون متشکل از وسایل نقلیه هوایی بدون سرنشین (UAV) ها، به عنوان مثال، هواپیماهای بدون سرنشین، و دیگر سنسورهای IoT هستند که با هم مجموعه‌ای از سنسورها، از جمله دوربین‌های فراطیفی، میکروفن‌ها، یا تبلت‌های غیر نظامی و گوشی‌های هوشمند را متصل می‌کنند. این سیستم‌ها در گذشته با موفقیت برای حمایت از اولین پاسخ دهندگان در سناریوهای انسانی یا بلایای طبیعی به کار گرفته شده‌اند. نقش هواپیماهای بدون سرنشین در IoT به‌طور کلی، و در پاسخ به فاجعه به طور خاص، می‌تواند در آینده برجسته‌تر شود زیرا آن‌ها پتانسیل فعال کردن، بهبود، و بهینه‌سازی عملیات و خدمات امداد و نجات جدید و موجود را دارند. هواپیماهای بدون سرنشین مستقل و نیمه مستقل بدون شک به‌طور گسترده‌تر در کمک به انسان‌ها در زمینه‌های دیگر ادامه خواهند داد، از بازرسی صنعتی گرفته تا بررسی عملیات و پشتیبانی عملیات نظامی.

یک شبکه از هواپیماهای بدون سرنشین را می توان برای جمع‌آوری مقادیر زیادی از داده‌ها به کار برد که پس از آن می‌توان آن‌ها را در لبه شبکه به صورت سنگین آپلود کرد. پردازش صوتی/ تصویری، که در آن منابع اجرای الگوریتم‌های یادگیری ماشینی (ML) در دسترس هستند.

در شرایط تحمیل‌شده توسط شبکه‌های به چالش کشیده شده مانند آن‌هایی که بعد از یک سناریوی بلایای طبیعی وجود دارند، حفظ عملکرد خوب این دستگاه‌های IoT می‌تواند یک چالش مهم باشد. اگرچه پروتکل‌ها و معماری‌های تحمل تاخیر و اختلال وجود دارند، مساله حفظ کیفیت قابل‌قبول خدمات با تاخیرهای شدید برای این شبکه‌ها، نه تنها به کیفیت اتصال بلکه به ماهیت پویای وظایفی که هواپیماهای بدون سرنشین برای انجام آن‌ها مورد نیاز هستند نیز بستگی دارد.

هم رویکردهای متمرکز و هم رویکردهای توزیع‌شده، که امکان ایجاد یک شبکه لبه‌ای از دستگاه‌های IoT، هواپیماهای بدون سرنشین، یا به طور کلی ربات‌ها، برای یک سرویس پایدار و قابل تطبیق را فراهم کنند، هم اکنون وجود دارند. برخی از آن‌ها بر روی مساله برنامه‌ریزی ماموریت انعطاف‌پذیر تمرکز می‌کنند و برخی دیگر بر روی راه‌حل‌های آگاهی از سلامت عوامل تمرکز می‌کنند. برخی دیگر بر روی مساله توانمندسازی تیم‌های چندعاملی برای مقابله اتوماتیک با ماموریت‌های پیچیده در مقیاس بزرگ، در طول دوره‌های زمانی طولانی در حضور شکست‌های محرک تمرکز کرده‌اند.

این راه‌حل‌ها طراحی صدا دارند، و مدل‌های شکست مختلف را تحت برنامه‌های کاربردی خاص مورد خطاب قرار می‌دهند، اما یک راه‌حل منحصر به فردی که از اجرای یک ماموریت انعطاف‌پذیر هواپیمای بدون سرنشین تحت تمام مدل‌های شکست ممکن و کاربردهای احتمالی تضمین کند، نمی‌تواند وجود داشته باشد. برای این منظور، نویسندگان مقاله یک معماری برای قابلیت برنامه‌ریزی شبکه‌های رباتیک APRON (Architecture Programmability of Robotic Networks) پیشنهاد کرده‌اند، و نتایج اولیه ارائه‌شده در (Ventrella et al., 2019) را با یک ارزیابی عمیق‌تر و یک مثال عملی از کاربرد AI برای UAV ها توسعه داده‌اند. این معماری قابلیت برنامه‌ریزی مکانیزم‌های مختلف درگیر در مساله اجرای ماموریت UAV ها یا دیگر عوامل توزیع‌شده مبتنی بر لبه را ممکن می‌سازد. APRON یک لایه نرم‌افزاری است که بین سیستم‌عامل (رباتیک) (به‌عنوان مثال ROS) و هر برنامه کاربردی نرم‌افزار IoT قرار می‌گیرد. معماری APRON شامل مکانیزم‌های مدیریت شبکه کلاسیک، مانند نظارت شبکه، تعمیر، و عملیات کنترل مانند کشف همسایه، و همچنین مکانیزم‌های خاص مساله اجرای ماموریت انعطاف‌پذیر است. در نهایت، یک رابط NorthBound را برای برنامه‌نویسان کاربردی فراهم می‌کند.

توضیح راه‌حل پیشنهادی مقاله برای حل مساله:

نویسندگان (در بخش ۲) این مقاله برخی از برنامه‌های کاربردی که در آن APRON می‌تواند مورد استفاده قرار گیرد را توصیف کرده‌اند. اگرچه با الهام از مورد استفاده از پاسخ به فاجعه، روش ایشان در واقع کاربرد گسترده‌تری دارد. شامل: پاسخ به بلایا و فاجعه، توسط سیستم هوایی بدون سرنشین (UAV) مانند هواپیماهای بدون سرنشین، و سیستم‌های حمل و نقل هوشمند و کاربرد اطلاعات و فناوری‌های ارتباطی (ICT) که در همه این‌ها می‌توان از معماری APRON بهره جست. نویسندگان مساله را در بخش ۳ ارائه کرده‌اند.

الف) مساله: با توجه به مجموعه‌ای از دستگاه‌ها که مجموعه‌ای از کارهای فشرده محاسباتی را بر روی یک سرور محاسباتی لبه تخلیه می‌کنند، به عنوان مثال، سیستم کنترل زمینی (GCS)، نویسندگان مساله برون‌ریزی وظیفه را به‌عنوان مساله مدیریت شبکه لبه تعریف کرده‌اند که میانگین زمان اتمام یک مجموعه از وظایف محول شده را توسط هماهنگ کردن موثر بار بر روی زیرساخت اساسی، به حداقل می‌رساند.

تنظیم بار محاسباتی لبه قابل‌برنامه‌ریزی شامل دو فرآیند اصلی است: (۱) اجرای پروفایل بار داده‌شده بر روی عوامل لبه، (۲) انتقال وظایفی که زمان اجرای مورد انتظار آن‌ها به طور قابل‌توجهی بالا است، به یک‌گره دیگر که به احتمال زیاد آن‌را در زمان کوتاه تری تکمیل می‌کند. اجرای پروفایل بار به این معنی است که راه‌حل در نظر دارد بار را در میان گره‌های زیرساخت طوری متعادل کند که گره‌های مشابه بارها هدف یکسانی داشته باشند. با این حال، تکنیک‌های تعادل بار زمانی که ممکن است شکست‌های شدید رخ دهند نامناسب هستند، از این رو نویسندگان فرآیند مهاجرت را با استفاده از یک مکانیزم خودتطبیقی طراحی کردند.

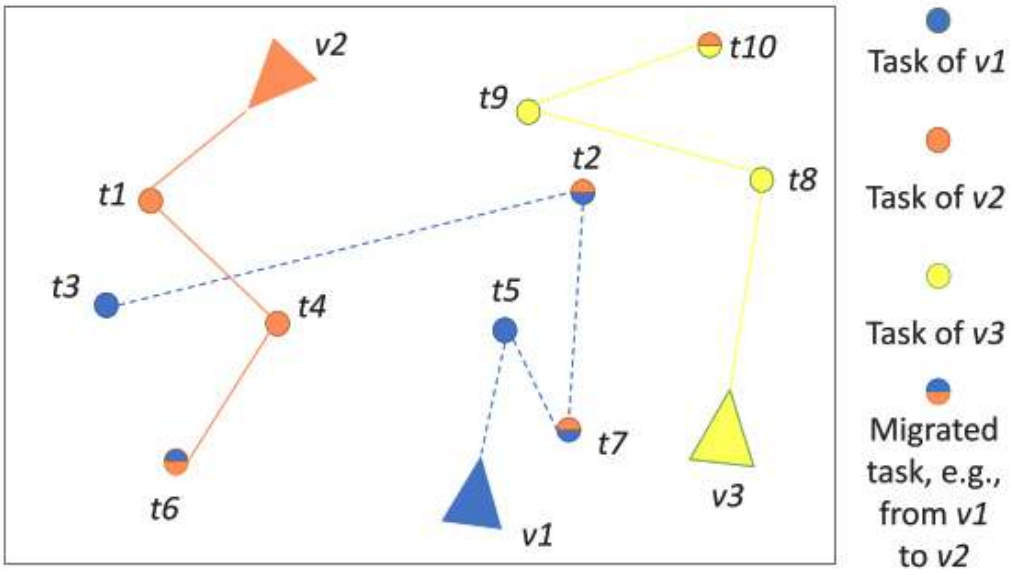
ب) خاصیت‌های مهاجرت وظیفه: برای حل مساله (الف)، باید به دو سوال توجه کرد: چه زمانی تخلیه مناسب است (هم از دستگاه تلفن همراه به GCS و هم در میان گره‌های تلفن همراه)، و وظیفه باید به کجا مهاجرت کند؟ راه‌حل ارائه‌شده از یک مکانیزم سازگاری خودتنظیم فعال استفاده می‌کند که وظیفه داده‌شده را هنگامی که گره میزبان (host) آن به یک آستانه می‌رسد، مهاجرت می‌دهد. چنین آستانه‌ای می‌تواند به دلیل معماری راه‌حل سفارشی شود (بخش ۵)، اما به‌عنوان پیش‌فرض، تعداد متوسط کارهای صف شده و در حال اجرا در کل سیستم لبه تنظیم می‌شود.

چه زمانی ما یک کار را به یک عامل مرزی دیگر مهاجرت می‌دهیم؟ هر عامل لبه با استفاده از مدل شبکه Jackson این آستانه را به صورت مستقل محاسبه می‌کند. از این مدل می‌توان برای برآورد تعداد متوسط وظایف در هر گره محاسباتی لبه i استفاده کرد که نشان‌دهنده چنین مقداری به صورت روبرو است. $\hat{n} = E[n_i]$

ما این شغل را به کجا منتقل می‌کنیم؟ گره مقصد طرح مهاجرت شغلی با مدل‌سازی شبکه به عنوان شبکه‌ای از صف‌ها انتخاب می‌شود. وظیفه می‌تواند از صف گره مبدا به صف گره مقصد مهاجرت کند. $P_{ii'}$ را احتمال مهاجرت بین منبع i و مقصد i' در نظر بگیرید. از نظر ریاضی استراتژی نویسندگان می‌تواند به‌صورت روبرو فرمول‌نویسی شود: در نظر بگیرید سیستم از گره‌های Q (صف‌ها) تشکیل شده باشد، گره مقصد dest که وظایف به آن محول می‌شوند، با حل معادله زیر انتخاب می‌شود:

$$dest = \arg\max_{i'} p_{ii'} \qquad (1)$$

پس از یک مرحله ارزیابی در مورد سربار تاخیر مهاجرت که با توجه به مدل ارائه‌شده در بخش ۴ محاسبه می‌شود، احتمال تضمین کارایی در روش تخلیه محاسبه می‌شود. شکل ۱ مثالی از برنامه کاری سه عامل را نشان می‌دهد. هر پهپاد مجموعه‌ای از وظایف را دریافت می‌کند و از یک مکان به مکان بعدی حرکت می‌کند که در آن باید کار را انجام دهد. به عنوان مثال، نماینده v_2 وظایف اصلی t_2 و t_7 خود را به پهپاد v_1 و t_{10} را به v_3 به‌منظور تعادل بار کلی انتقال می‌دهد.



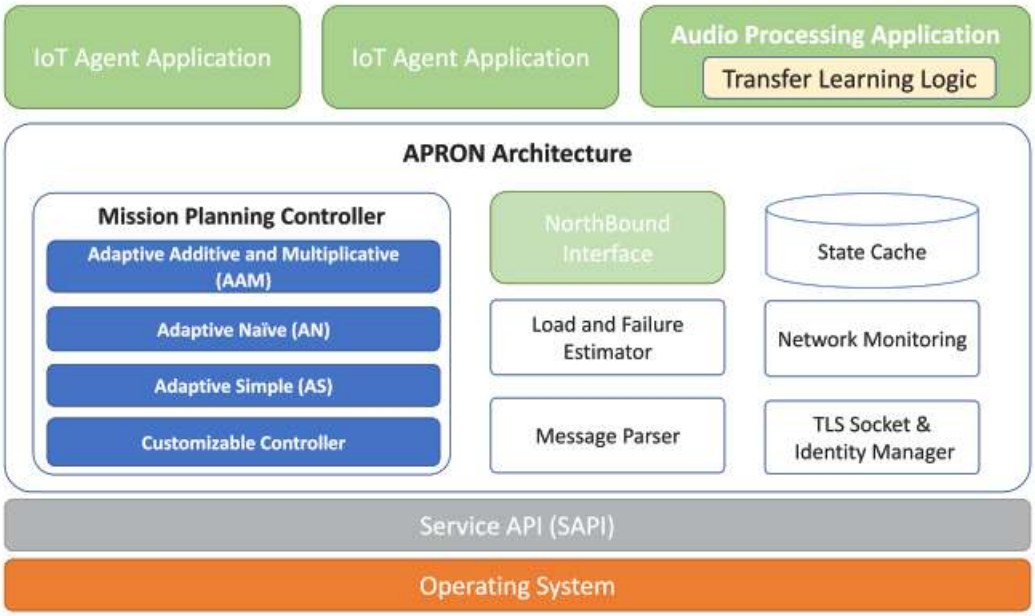
شکل ۱- نمونه ای از مسیرهای برنامه ریزی مأموریت که توسط سه هواپیمای بدون سرنشین دنبال می‌شوند تا وظایف خود را تکمیل کنند. وظایف مهاجرت در بین گره ها رخ می دهد تا بار کلی را متعادل کند.
نویسندگان از این شکل فقط با چند هواپیمای بدون سرنشین برای روشن کردن روند مهاجرت و گردش کار استفاده کردند، اما ایشان مقیاس پذیری APRON را در تجربیات بعدی آزمایش کردند.

سپس نویسندگان به تفصیل جزء تخمین زننده بار و خطا را توصیف کرده‌اند (بخش ۴ مقاله)، که از یک مدل شبکه جکسون برای پشتیبانی از نظارت و کنترل عملیات در حالی که حالت‌های شبکه تکامل می‌یابند، استفاده می‌کند. تخمین‌گر یک شکل نزدیک به متوسط تعداد وظایف در یک ماموریت را محاسبه می‌کند، چه در صف باشند و چه در اجرا. چنین برآوردکننده‌ای می‌تواند توسط برنامه‌نویسان نرم‌افزار برای تعیین استفاده از هر یک از هواپیماهای بدون سرنشین و میانگین زمان صف (هم زمان انتظار و هم زمان اجرا) برای هر کار دستکاری شود. بنابراین چنین اطلاعاتی می‌توانند برای طراحی کنترل‌کننده‌هایی که با برنامه‌های خاص سازگار می‌شوند (همراه با APRON API این مقاله) مورد استفاده قرار گیرند.

معماری APRON پیشنهادی: در این بخش، نویسندگان لایه مدیریتی پیشنهادی خود را در میان سیستم‌عامل (در پایین)، به عنوان مثال، سیستم‌عامل رباتیک (ROS)، و برنامه کاربردی IoT (در بالا) ارائه داده‌اند. شکل ۲ ساختار مدیریتی را نشان می‌دهد که مکانیسم‌های آن امکان ایجاد و نظارت بر اتصال شبکه، تخمین خرابی گره و لینک و برنامه‌ریزی مجدد ماموریت از طریق یک منطق کنترل‌کننده قابل تنظیم را فراهم می‌آورد. برنامه در بالا می‌تواند از API ارائه‌شده برای سفارشی کردن منطق این کنترل‌کننده‌ها، تطبیق با مدل‌های شکست مختلف، و همچنین سفارشی کردن منطق برنامه‌ریزی ماموریت، در یک مد متمرکز یا

توزیع‌شده، استفاده کند. در ادامه، اجزای شبکه راه‌حل، از جمله API، و خدمات نماینده ماموریت ارائه‌شده را به‌صورت خلاصه آورده شده است.

۱) نظارت شبکه (Network Monitoring): با الهام از اکثریت قریب به اتفاق سیستم‌های شبکه‌ای، مولفه مدیریت اتصال یک پروتکل کشف شبکه را اجرا می‌کند، و یک فرآیند سگ نگهبان یک پروتکل ضربان قلب را برای نظارت بر ارتباطات زنده اجرا می‌کند. معماری به یک آدرس IP نیاز ندارد و از این رو، کاستی‌های چندگانه و تحرک معماری TCP / IP را به ارث نمی‌برد. همانند ساختارهای تمیز و پاک شده اینترنت که اخیرا پیشنهاد شده‌اند (Day et al., 2008)، نویسندگان آدرس‌های عامل را به نام‌های برنامه متصل می‌کنند، نه به واسطهای شبکه.



شکل ۲- معماری APRON: یک لایه مدیریتی بین برنامه IoT و سیستم عامل برای ایجاد و نظارت بر اتصال شبکه، برای تخمین خرابی‌ها و برای انطباق برنامه ریزی (دوباره) کار بر اساس منطق کنترل‌کننده قابل تنظیم است.

۲) سوکت TLS و مدیر هویت: از آنجا که هر نماینده ممکن است به چندین تداخل تعلق داشته باشد، لازم است که پیش از برقراری ارتباط تصدیق شود. این جز مسئول مدیریت هویت عامل در سراسر شبکه‌های هم پوشانی چندگانه است و از طریق پروتکل امنیت لایه حمل و نقل (TLS) یک اتصال امن را فراهم می‌کند.

۳) مدیر حفظ حالت: این جز در عین حفظ حالات شبکه، پایگاه‌داده نسبتا تکراری را کنترل می‌کند. مدخل‌های پایگاه‌داده یا مربوط به حالت‌های ایستا می‌باشند، مانند حالت صرفا وابسته به عامل، یا مربوط به حالت‌های

دینامیکی هستند، حالت‌هایی که به شبکه، پیکربندی و شرایط اتصال وابسته هستند. حفظ حالت همچنین به‌عنوان یک لگاریتم برای ذخیره حالت‌های برنامه کاربردی، برای مثال استفاده از باتری دستگاه IoT ، نیز استفاده می‌شود.

۴) سرویس API: این سرویس از سفارشی سازی دو مولفه اصلی پشتیبانی می‌کند: (i) منطق کنترل‌کننده، که می‌تواند با سناریوهای چندگانه (شکست) تناسب داشته باشد، (ii) منطق الگوریتم برنامه‌ریزی ماموریت، چه به‌صورت متمرکز و چه به‌صورت توزیع‌شده. به این ترتیب، برنامه‌ی مشابه آن زمینه‌های متفاوت، انطباق با الزامات مختلف و شرایط شبکه را دنبال می‌کند.

۵) پیغام تجزیه‌گر و مدل شیء: برای تعریف مدل هدف مقاله، و همچنین برای پیاده‌سازی منطق تعیین کردن، ردیف کردن و حذف پیغام، نویسندگان از پروتکل بافر (Buffer) گوگل استفاده کرده‌اند، زیرا از دیگر زبان‌های نمادسازی نحوی مبتنی بر متن مانند Json و XML کارآمدتر است.

۶) برنامه IoT: در بالای معماری APRON، برنامه کاربردی می‌تواند از خدمات ارائه‌شده توسط معماری بهره‌برداری کند. نمونه‌هایی از این کاربردها قبلا (در بخش ۲) ارائه شده‌اند. علاوه بر آن این مقاله یک کاربرد تجزیه و تحلیل صوتی زنده با بیان مفهوم یادگیری انتقالی، ارائه شده است.

۷) رابط NorthBound: این برنامه می‌تواند با معماری APRON از طریق API هایی که رابط NorthBound را تشکیل می‌دهند، ارتباط برقرار کند. تمام خدمات ارائه‌شده در واقع از طریق REST API ها، یک استاندارد غیر عملی، که منابع را با استفاده از یک مجموعه یکنواخت و از پیش تعریف‌شده از عملیات بدون تابعیت، در معرض قرار می‌دهد، در دسترس هستند.

۸) برآورد شکست و بار: این جز هسته اصلی معماری است زیرا اطلاعات مورد نیاز برای انجام مهاجرت وظیفه را فراهم می‌کند. با متد ارائه شده در بخش ۴ مقاله، آن تعداد متوسط وظایف مورد انتظار را محاسبه می‌کند و حالت فعلی را برای تخمین نرخ شکست مقایسه می‌کند. بار به صورت فعالانه توزیع می‌شود تا از شکست جلوگیری شود، و وقتی شکست رخ می‌دهد، وظایف قبلی در صف شکست‌خورده به یک عامل دیگر منتقل می‌شوند.

۹) کنترل‌کننده برنامه‌ریزی ماموریت: این چارچوب از یک کلاس از کنترل‌کننده‌ها برای تناسب نرخ برنامه‌ریزی مجدد ماموریت R(t) شبکه دستگاه‌های IoT، به عنوان مثال، هواپیماهای بدون سرنشین، پشتیبانی می‌کند. با این حال، معماری این مقاله مدولار و قابل اتصال است، از این رو می‌تواند با دیگر کنترلرهای تعریف‌شده توسط کاربر گسترش یابد. منظور نویسندگان از کنترلر، کنترلر شبکه‌سازی تعریف‌شده توسط نرم‌افزار (SDN) نیست، بلکه کنترلر بازخورد است. ما در حال حاضر برخی از کنترلرها را اجرا کرده‌ایم که نرخ آن‌ها به $\hat{n}(t)$ بستگی دارد که تعداد تخمینی وظایف در حال حاضر در شبکه را نشان می‌دهد و قبلا در بخش ۴ بحث شده‌است.

۱۰) کنترل‌کننده ساده انطباقی (AN): نرخ برنامه‌ریزی مجدد با نسبت بین $\hat{n}(t)$ و $\acute{n}$ تغییر می‌کند که نشان‌دهنده تعداد وظایف مطلوب در سیستم در حالت پایدار براساس معادله زیر است:

$$R(t+1) = \frac{\hat{n}(t)}{\acute{n}} R(t)$$

۱۱) کنترل‌کننده ساده انطباقی (AS): نرخ برنامه‌ریزی مجدد با $\hat{n}(t)$ و $\acute{n}$ با توجه به معادله زیر تغییر می‌کند:

$$R(t+1) = k(\hat{n}(t) - \acute{n})$$

که در آن k یک ثابت مثبت است.

۱۲) کنترل‌کننده تطبیقی افزودنی و چند ضربی (AMM): نرخ برنامه‌ریزی مجدد با $\acute{n}$ و $\acute{m}$ که به ترتیب تعداد کارهای تکمیل‌شده و تعداد تخمینی وظایف از دست رفته در زمان t تغییر می‌کند:

$$R(t+1) = \begin{cases} k\frac{\hat{m}(t)}{\tilde{n}(t)+1} & \hat{m}/(\tilde{n}+1) > 0 \\ -\alpha & \text{otherwise} \end{cases}$$

که در آن α یک ثابت مثبت است.

۱۲) کنترل‌کننده قابل تنظیم: جدا از استراتژی‌های ارائه‌شده که معماری با آن‌ها تجهیز شده‌است، نرخ برنامه‌ریزی مجدد می‌تواند سیاست‌های دیگر تعریف‌شده توسط کاربر، برای مثال از طریق رابط NorthBound را دنبال کند.

نویسندگان اجزای معماری APRON خود را در بخش ۵ مقاله با تمرکز بر جز کنترل‌کننده آن و چگونگی دستیابی به قابلیت برنامه‌ریزی مرور می‌کنند. کاربرد پردازش صوتی در بخش ۶ مقاله ارائه شده‌است، که جزئیات مربوط به مدل DNN های پیشرفته ما را ارائه می‌دهد. نویسندگان نتایج تجربی خود را در بخش ۷ ارائه داده‌اند، در حالی که ارائه مقالات در بخش ۸ و بخش ۹ مقاله ایشان را نتیجه‌گیری می‌کند.

نتایج ارزیابی: نویسندگان عملکرد راه‌حل خود را با توسعه یک شبیه‌ساز رویداد محور C++ که قادر به اجرا در هر ماشین است، ارزیابی کرده‌اند. در طول این ارزیابی، نویسندگان یک شبکه از ناوگان هواپیماهای بدون سرنشین مستقر شده برای انجام یک ماموریت، متشکل از مجموعه‌ای از اقدامات سفارش داده‌شده توسط GCS را در نظر گرفتند. نمونه‌هایی از این اقدامات می‌تواند موقعیت‌های جغرافیایی برای دسترسی به منظور کشف منطقه از طریق دوربین و میکروفون باشد، ارسال جریان‌هایی که برای تعیین محل بازماندگان در پاسخ

به فاجعه شرح داده خواهد شد. هر یک از هواپیماهای بدون سرنشین دستورالعمل‌ها را دریافت می‌کند و سعی می‌کند آن‌ها را اجرا کند. در عین حال مشخص می‌شود که آیا مهاجرت با محاسبه آستانه ای ضروری است یا خیر. این مقدار با استفاده از مدل شبکه جکسون نویسندگان تخمین زده می‌شود و باعث مهاجرت وظایف در تابلو می‌شود. برای این منظور، تمام عوامل برای تکمیل کارهای اختصاص‌داده‌شده در کوتاه‌ترین زمان ممکن هم‌کاری می‌کنند.

نقاط قوت و ضعف مقاله

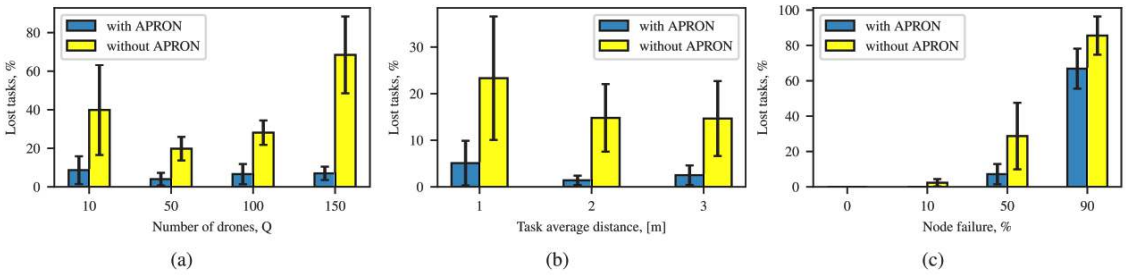
نویسندگان مقاله دو کار عمده انجام داده‌اند. اول اینکه معماری میان‌افزار خود را برای مدیریت دستگاه IoT به تفصیل شرح داده‌اند و یک الگوریتم بهینه‌سازی را برای برنامه‌ریزی مجدد کار، در مورد یک دستگاه IoT که یک شکست را تجربه می‌کند، ارائه داده‌اند. دوم اینکه ایشان یک برنامه کاربردی پاسخ به فاجعه را به‌عنوان کاربرد مورد استفاده AI معرفی کرده‌اند که از خدمات شبکه انعطاف‌پذیر اساسی APRON استفاده می‌کند. برنامه کاربردی ایشان برای شناسایی صداهای تولید شده توسط انسان‌ها طراحی شده‌است، برای مثال، صداهای قربانیان یک فاجعه طبیعی یا انسانی که باید نجات داده شوند، یا بازماندگان قرار گرفته در زیر بهمن، جایی که ممکن است ویدیو به تنهایی کافی نباشد. به طور خاص، برنامه پردازش صوتی لبه این مقاله از تکنیک‌های شبکه‌های عصبی عمیق (DNN ها) برای طبقه‌بندی صوت ارسالی از ناوگان هواپیماهای بدون سرنشین استفاده می‌کند و به تعیین موقعیت صدای انسان کمک می‌کند. از آنجاییکه سرعت برای عملیات‌های نجات بسیار مهم است، نویسندگان مقاله به بررسی و بهره‌برداری از ویژگی‌های یادگیری انتقالی (Transfer Learning) به منظور کاهش زمان آموزش و افزایش دقت طبقه‌بندی (صدای انسان/ غیر انسان) پرداخته‌اند.

معماری ارائه شده در مقاله به یک آدرس IP نیاز ندارد و از این رو، کاستی‌های چندگانه و تحرک معماری TCP / IP را به ارث نمی‌برد. همچنین معماری این مقاله مدولار و قابل اتصال است، از این رو می‌تواند با دیگر کنترلرهای تعریف‌شده توسط کاربر گسترش یابد. منظور نویسندگان از کنترلر، کنترلر شبکه‌سازی تعریف‌شده توسط نرم‌افزار (SDN) نیست، بلکه کنترلر بازخورد است.

نویسندگان مقاله همچنین از شبیه‌ساز تخصیص ماموریت برای تست مقیاس‌پذیری روش خود استفاده کرده-اند، و راه‌حل‌های خود را بر روی یک نمونه اولیه طراحی‌شده توسط شبکه آزمایشی مجازی برای ارزیابی عملی بودن APRON گسترش داده‌اند.

مزیت‌های کاربردی: علاوه بر برآورد عملکرد سیستم و مقایسه سیاست‌های مختلف APRON، نویسندگان مزایای ملموس را برای یک کاربرد محاسباتی لبه ارزیابی کرده‌اند. نویسندگان به طور خاص کاربرد پیشنهادی (بخش ۶) را در مواردی که APRON مستقر باشد یا نباشد، آزمایش کرده‌اند. در این سناریو، هواپیماهای بدون سرنشین وظیفه رسیدن به موقعیت جغرافیایی مانند مثال‌های قبلی و وظیفه ضبط صدا در پس‌زمینه را انجام می‌دهند. شرایط نامطلوب شبکه‌های به چالش کشیده شده برای مواجهه با وظایف از دست رفته تحمیل می‌شود زیرا گره‌ای که آن‌ها را میزبانی می‌کرد با شکست مواجه شد. سیستم قادر به تخصیص مجدد این

وظایف است، اما تاخیر درک شده توسط کاربر به شدت افزایش می‌یابد. به همین دلیل، لایه‌ای مانند APRON در کاهش اثرات ناشی از شکست موثر است. شکل ۳ دو مزیت اصلی APRON را نشان می‌دهد: (۱)پاسخ خطای موثر، (۲)تخمین خطای دقیق؛ لایه مدیریتی ارائه‌شده توسط APRON اجازه تعداد کمتری از وظایف از دست رفته در سناریوهای مختلف را می‌دهد. حضور APRON به ویژه در شرایط بحرانی، یعنی درصد بالای شکست گره، تعداد زیادی از هواپیماهای بدون سرنشین برای کنترل، مشهود است. از سوی دیگر، فاصله بین گره‌ها به طور قابل‌توجهی بر تعداد کاره‌ای تکمیل‌شده تاثیر نمی‌گذارد.



شکل ۳- مقایسه عملکرد برنامه با استفاده از نزدیکترین سیاستهای برنامه ریزی مهاجرت وظیفه. نمودارها نشان‌دهنده درصد وظایف از دست رفته هنگام انجام ۵۰۰ کار است در شرایط مختلف: (a) تعداد هواپیماهای بدون سرنشین ، (b) فاصله کار ، (c) درصد خرابی گره.

جمع‌بندی و پیشنهادات برای کارهای آتی

گسترش اینترنت اشیا (IoT) در ارتباط با محاسبات لبه اخیرا چندین احتمال را برای چندین برنامه جدید مطرح کرده است. نمونه‌های معمول، وسایل نقلیه هوایی بدون سرنشین (UAV) هستند که برای پاسخ سریع به فاجعه، فتوگرامتری، نظارت و نظارت محیطی مستقر شده‌اند. برای پشتیبانی از پیشرفت رو به رشد برنامه‌های کاربردی به کمک یادگیری ماشینی در همه این برنامه‌های کاربردی شبکه‌ای، یک چالش رایج ارائه یک سرویس پایدار و مداوم است، یعنی سرویسی که قادر به حفظ مداوم سطح بالایی از عملکرد در مواجهه با شکست‌های احتمالی باشد. برای پرداختن به این چالش‌های انعطاف‌پذیر خدمات، نویسندگان APRON را که یک راه‌حل حاشیه‌ای برای مدیریت برنامه‌ریزی کار توزیع‌شده و سازگار در یک شبکه از دستگاه‌های IoT است، به‌عنوان مثال هواپیماهای بدون سرنشین را پیشنهاد کرده‌اند. با استفاده از مدل شبکه جکسون، این معماری مقاله حاضر یک استراتژی برنامه‌ریزی جدید را برای پشتیبانی بهتر از عملیات کنترل و نظارت در حالی که حالت‌های شبکه تکامل می‌یابد، به‌کار می‌برد. برای نشان دادن قابلیت‌های معماری پیشنهادی، نویسندگان همچنین یک برنامه تشخیص صدا مبتنی بر یادگیری عمیق را با استفاده از رابطه APRON

NorthBound، برای تشخیص صدای انسان در شبکه‌های به چالش کشیده شده اجرا کرده‌اند. منطق برنامه از یادگیری انتقال (Transfer Learning) برای بهبود دقت طبقه‌بندی صدا و زمان اجرای عملیات نجات مبتنی بر UAV استفاده می‌کند.

این مقاله یک کاربرد محاسباتی لبه جدید را ارائه می‌دهد که با استفاده از الگوریتم های یادگیری ماشینی، قادر به تشخیص حضور انسان در سناریوهای فاجعه است. برای سرعت بخشیدن به محاسبات و تضمین قابلیت اطمینان قابل‌قبول برنامه، نویسندگان همچنین یک معماری مدیریت را گسترش داده‌اند که هدف آن برنامه‌ریزی مجدد وظایف در حضور شبکه‌های لبه به چالش کشیده شده است. چنین لایه‌ای از مدل صف شبکه جکسون برای تخمین تعداد وظایف، صف و یا در اجرا استفاده می‌کند. بنابراین، برنامه کاربردی می‌تواند کاربرد آنی هر دستگاه IoT، و میانگین زمان صف (هم زمان انتظار و هم زمان اجرا) برای هر وظیفه اجرا شده یا محول شده به لبه شبکه را تعیین کند. نتایج ایشان نشان می‌دهد که چگونه این لایه مدیریت یک ابزار موثر برای قابلیت برنامه‌ریزی سیاست مساله برنامه‌ریزی مجدد ماموریت برای هر دستگاه IoT مستقر در محیط‌های شبکه‌بندی شده به چالش کشیده شده است. علاوه بر این، زمان پردازش صدا زمانی کاهش می‌یابد که سرویس اصلی در حال اجرا است، زیرا برنامه در بالای آن می‌تواند از ویژگی‌هایی بهره‌برداری کند که قادر به بهبود عملکرد کلی سیستم باشند.

در این مقاله بطور مشخص پیشنهادی برای تحقیقات آتی ارائه نشده است.

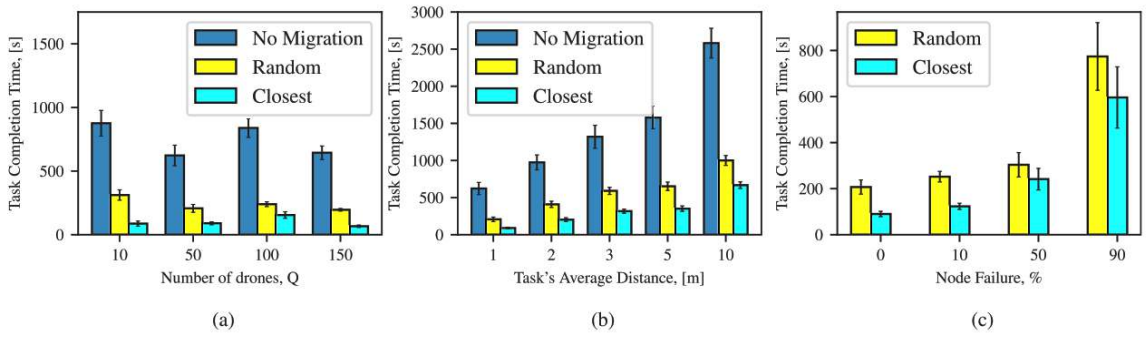
شبیه سازی

نویسندگان در جدول ۱ پارامترهای پیکربندی مورد استفاده در طول ارزیابی زیر را به طور خلاصه نشان می‌دهیم، که در آن مقادیر پیش‌فرض به صورت برجسته گزارش شده‌اند.

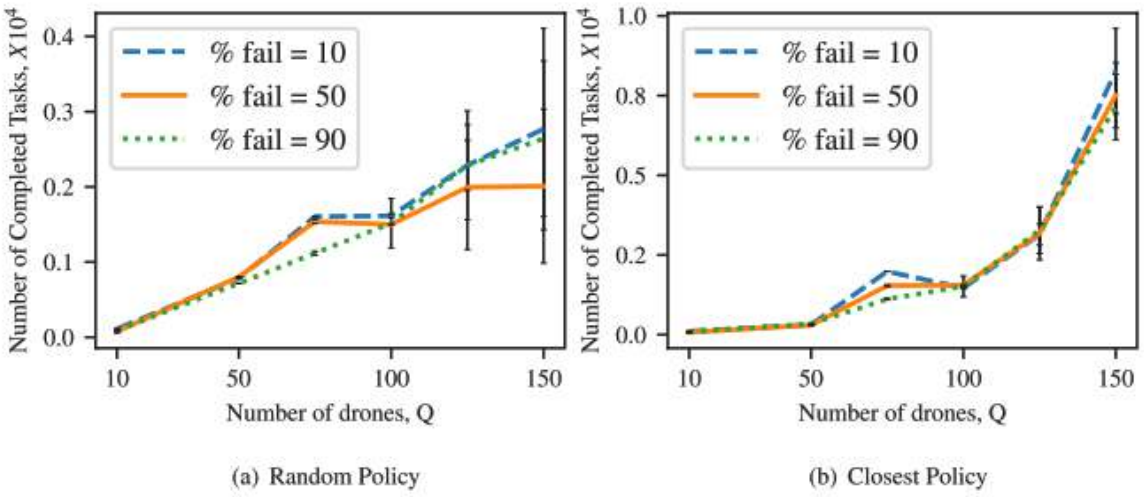
جدول ۱- تنظیمات پارامترها

Parameter	Values
Number of nodes	10, **50**, 100, 150
Nodes' Average Distance [m]	**1**, 2, 3, 5, 10
Node failure [%]	**0**, 10, 50, 90
Number of Trials	**30**
Confidence Interval [%]	**90**

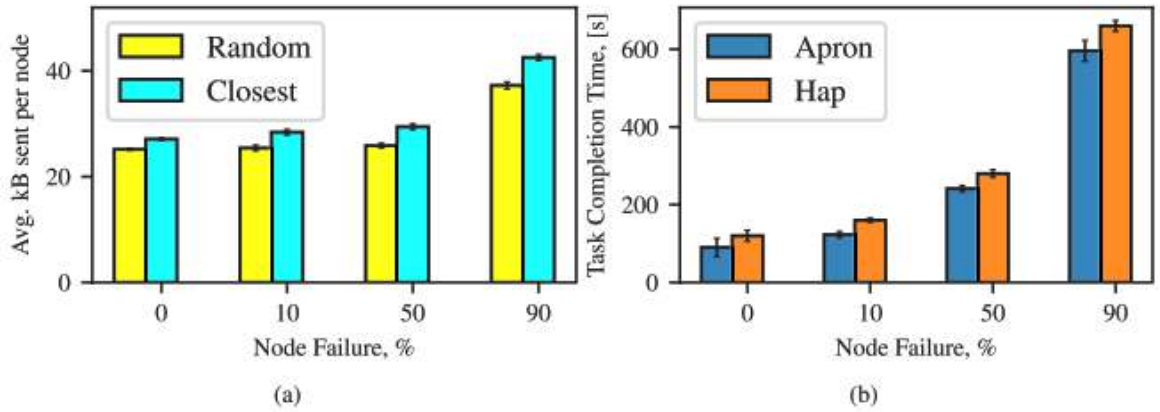
نویسندگان آزمایشاتی را بر روی ناوگان با اندازه‌های مختلف اجرا کرده‌اند که شامل تعداد هواپیماهای بدون سرنشین که ۱۰، ۵۰، ۱۰۰ یا ۱۵۰ است. تست‌ها همچنین تغییر میانگین فاصله بین دو ژئوموقعیت متوالی باید توسط یک پهپاد ۱، ۲، ۳، ۵ یا ۱۰ متر مورد بازدید قرار گیرد، را در نظر می‌گیرند. علاوه بر این، نویسندگان عملکرد را در مورد سه سیاست مهاجرت وظیفه مجزا ارزیابی کرده‌اند: (۱) بدون برنامه‌ریزی مجدد (مهاجرت وظیفه): عوامل در سیستم هم‌کاری نمی‌کنند، اما هر یک از آن‌ها تلاش می‌کند تا همه و تنها وظایف را در صف خود انجام دهد؛ (۲) برنامه‌ریزی مجدد وظیفه تصادفی: هنگامی که صف یک نماینده از یک آستانه تعیین‌شده تجاوز می‌کند، هواپیمای بدون سرنشین بعدی که وظایف را به صورت مازاد دریافت خواهد کرد، به طور تصادفی انتخاب می‌شود؛ (۳) نزدیک‌ترین برنامه‌ریزی مجدد وظیفه: هنگامی که صف یک نماینده بر آستانه غلبه می‌کند، سیستم وظایف خود را به نزدیک‌ترین گره تخصیص می‌دهد. در صورتیکه دو یا چند عامل در فاصله یکسان از کار وجود داشته باشد، گره مقصد، گره با وظایف کم‌تر در صف خود است؛ در صورتی که دو صف دارای تعداد وظایف یک‌سان باشند، گره‌ها به صورت تصادفی تقسیم می‌شوند.



شکل ۴- زمان اتمام کار ناوگان هواپیماهای بدون سرنشین با استفاده از APRON با سیاست‌های مختلف برنامه‌ریزی: (i) عدم انتقال کار، (ii) مهاجرت وظیفه تصادفی، (iii) نزدیکترین مهاجرت کار. نمودارها زمان اتمام کار را در شرایط مختلف نشان می‌دهند: (a) تعداد هواپیماهای بدون سرنشین ، (b) فاصله کار ، (c) درصد خرابی گره ها.



شکل ۵- نمودارهای استقامت، یعنی تعداد کارهای انجام شده قبل از اولین شکست، برای (a) سیاست تصادفی ، و (b) نزدیکترین سیاست.



شکل ۶- (a) بایت ها در هر گره برای افزایش تعداد عوامل در شبکه رد و بدل می‌شوند. (b) مقایسه با معماری‌های مختلف، از نظر زمان برای انجام وظایف.

نتایج نشان می‌دهد که چگونه این چارچوب یک ابزار موثر برای مساله تخصیص مجدد مبتنی بر سیاست است. چند مشاهده را می توان از شکل‌ها ۴-۵-۶ با توجه به عملکرد کلی سیستم استنباط شود:

(۱) سیاست‌های مهاجرت وظیفه زمان اتمام ماموریت کوتاه‌تری را نشان می‌دهند. همانطور که در شکل 5a دیده می‌شود اجرای سیاست تخصیص مجدد به عوامل اجازه می‌دهد تا وظایف خود را در زمان کوتاه تری به اتمام برسانند.

(۲) نزدیک‌ترین سیاست عامل زمان اتمام کمتری را با توجه به سیاست برنامه‌ریزی مجدد وظیفه تصادفی به دست می‌آورد.

(۳) زمان تکمیل کار با فاصله سفر نماینده کاهش می‌یابد.

(۴) زمان تکمیل با افزایش شکست هواپیماهای بدون سرنشین افزایش می‌یابد.

(۵) وقتی تعداد هواپیماهای بدون سرنشین به طور منطقی کم باشد، تعداد شکست‌ها بر عملکرد تاثیر نمی‌گذارد.

(۶) بهبود عملکرد نزدیک‌ترین سیاست شامل مقدار پیام‌های مبادله‌شده بزرگ‌تر و در عین حال معقول است.