Journal of Soft Computing and Information Technology (JSCIT)
Babol Noshirvani University of Technology, Babol, Iran
Journal Homepage: jscit.nit.ac.ir
Volume 13, Number 3, Autumn 2024, pp. 12-30
Received: 04/07/2024, Revised: 12/10/2024, Accepted: 12/19/2024

The Impact of Software-Defined Networks on Reducing Latency and Enhancing Data Security in Blockchain-Based IoT Architectures

Afsaneh Banitalebi Dehkordi[1]
1*- Department of Computer Engineering, Payame Noor University (PNU), P. O. Box 19395-4697, Tehran, Iran
[1 *]abanitalebi@pnu.ac.ir

Corresponding author's address: Afsaneh Banitalebi Dehkordi, Faculty of Computer Engineering, Department of Computer Engineering, Payame Noor University (PNU), P. O. Box 19395-4697, Tehran, Iran.

Abstract- The Internet of Things (IoT) is a network of objects interconnected via the Internet, enabling communication and information exchange among them to enhance the quality of human life. Despite its numerous advantages, IoT still faces challenges such as data security, privacy concerns, and high energy consumption of sensors. In this paper, a novel optimized hybrid architecture named "BlockC-SDOT" is proposed, leveraging blockchain and software-defined networking (SDN) architectures on a cloud computing platform for IoT. This innovative architecture combines a blockchain framework based on the Proof of Stake (PoS) consensus algorithm with SDN and cloud computing to enhance IoT performance and security. Through network clustering and cluster management via SDN, the architecture simultaneously addresses two critical parameters: sensor energy consumption and latency reduction in IoT networks. For enhanced security, the Diffie-Hellman algorithm and digital signatures are employed. Optimization is achieved using the Whale Optimization Algorithm (WOA) and the Jackson queueing model .To evaluate the efficiency of the proposed architecture, simulations are conducted to analyze various parameters, including latency, CPU performance under different attacks, throughput, energy consumption, packet arrival rate, and network bandwidth. These parameters are computed using the proposed method and compared against the results of the Ant Colony Optimization (ACO) and Genetic Algorithm (GA). Additionally, the latency of the proposed architecture is compared with the results obtained from RPL and CARP routing protocols. In both areas, the findings demonstrate the superiority and higher efficiency of the proposed hybrid approach over existing methods in the field.

Keywords- Internet of Thing, Energy optimization, Security, Blockchain, Software Defined Network.

مجله علمی رایانش نرم و فناوری اطلاعات
دانشگاه صنعتی نوشیروانی بابل
صفحه مجله: jscit.nit.ac.ir
جلد ۱۳، شماره ۳، پائیز ۱۴۰۳، صفحه ۱۲-۳۰
دریافت: ۱۴۰۳/۰۱/۱۹، بازنگری: ۱۴۰۳/۰۹/۲۰، پذیرش: ۱۴۰۳/۰۹/۲۹

بررسی تأثیر شبکه‌های نرم‌افزارمحور در کاهش تأخیر و بهبود امنیت داده در معماری بلاکچین‌محور اینترنت اشیاء

افسانه بنی طالبی دهکردی*[1]
*۱- استادیار، گروه مهندسی کامپیوتر، دانشگاه پیام نور، ص. پ. ۴۶۹۷-۱۹۳۹۵، تهران، ایران.
[1*]abanitalebi@pnu.ac. ir

* نشانی نویسنده مسئول: افسانه بنی طالبی دهکردی، دانشگاه پیام نور، ص. پ. ۴۶۹۷-۱۹۳۹۵، تهران، ایران.

چکیده- اینترنت اشیا، شبکه‌ای از اشیا بوده که از طریق اینترنت به یکدیگر متصل می‌شوند. این فناوری، قابلیت ارتباط و تبادل اطلاعات را بین اشیا فراهم می‌آورد وسبب بهبود کیفیت زندگی افراد می‌شود. با این حال، در حالی که اینترنت اشیا از مزایای بسیاری برخوردار است، مسائلی همچون امنیت و حریم خصوصی اطلاعات و مصرف انرژی بالای حسگرها، همچنان یک موضوع مورد بحث دراین شبکه‌ها است. دراین مقاله، یک معماری بهینه ترکیبی نوین با استفاده از معماری بلاک‌چین و شبکه‌های نرم‌افزارمحور با نام "BlockC-SDOT" دریک سکوی رایانش ابری برای اینترنت اشیا ارایه شده است. این معماری نوین از ترکیب معماری بلاک‌چین مبتنی بر الگوریتم اجماع اثبات سهام و شبکه‌های نرم‌افزار محور با محاسبات ابری برای بهبود عملکرد و ایمنی اینترنت اشیا استفاده می‌کند. دراین معماری با استفاده از خوشه‌بندی شبکه و مدیریت کنترل هر خوشه توسط شبکه‌های نرم‌افزار محور، به صورت همزمان به هردو پارامتر میزان مصرف انرژی حسگرها و میزان کاهش تاخیر در شبکه اینترنت اشیا توجه شده است. برای افزایش امنیت، از الگوریتم دیفی‌هلمن و امضاهای دیجیتال استفاده شده‌است. برای بهینه‌سازی معماری، الگوریتم فراابتکاری نهنگ و الگوریتم صف‌بندی جکسون مورد استفاده قرارگرفته‌اند. برای ارزیابی کارایی معماری پیشنهادی، شبیه‌سازی‌هایی انجام شده است که به تحلیل پارامترهای مختلفی از جمله میزان تاخیر، عملکرد سی‌پی‌یو در حملات مختلف، توان‌عملیاتی، مصرف انرژی، نرخ ورود بسته‌ها، و پهنای‌باند در شبکه مورد نظر پرداخته است. این پارامترها با استفاده از روش پیشنهادی محاسبه ونتایج بدست آمده با نتایج حاصل از الگوریتم کلونی مورچگان (Ant Colony Optimization)، و الگوریتم ژنتیک (Genetic Algorithm)، مورد مقایسه قرار گرفته‌اند. به‌علاوه، میزان تاخیر در معماری پیشنهادی با نتایج حاصل از پروتکل‌های مسیریابی RPL(Routing Protocol) و CARP(Channel-aware Routing Protocol) مورد بررسی و مقایسه قرار گرفته اند.درهردوزمینه، نتایج به دست آمده نشان دهنده برتری و کارآیی بیشتر رویکرد ترکیبی پیشنهادی نسبت به روش‌های موجود در این زمینه است.

واژه‌های‌کلیدی: اینترنت اشیا، بهینه‌سازی انرژی، امنیت، بلاک‌چین، شبکه‌های نرم‌افزار محور

۱- مقدمه

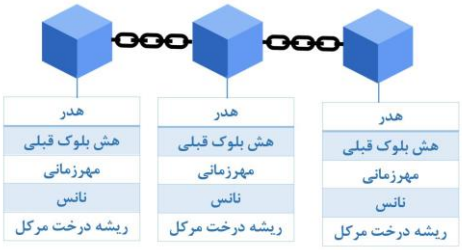
دراینترنت اشیا[1]، مجموعه‌ای از اشیا ودستگاه‌ها از طریق اتصال به یک شبکه، با یکدیگر ارتباط برقرار می‌کنند. درپژوهش‌های انجام شده دراین حوزه، اتصال حدود ۳۰ میلیارد دستگاه از طریق اینترنت و افزایش این تعداد به بیش از ۷۵ میلیارد دستگاه تا سال ۲۰۲۵ گزارش شده‌است[۱]. شکل ۱ کاربردهای اینترنت اشیا درصنایع مختلف را نشان می‌دهد.

گسترش وافزایش تعداد دستگاه‌ها در اینترنت اشیا، نیازمند افزایش ایمنی اطلاعات منتقل شده و بررسی میزان انرژی مصرف شده در کل شبکه وحسگرها[۲] است. دراین شبکه به دلیل تولید حجم زیادی از داده‌ها توسط حسگرها، مقدارانرژی زیادی مصرف می‌شود[۲].



شکل ۱: اینترنت اشیا

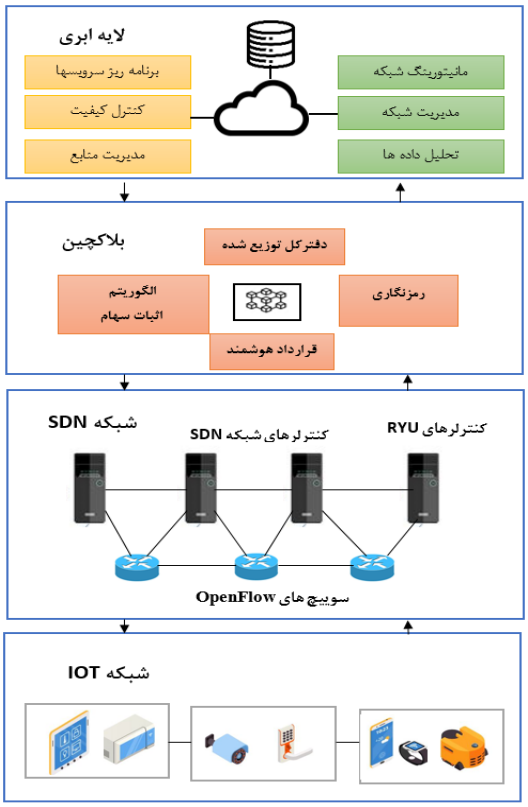
دراین مقاله یک معماری بهینه ترکیبی نوین با استفاده از معماری بلاک‌چین[۳] وشبکه‌های نرم‌افزار محور[۴] با نام"BlockC-SDOT" دریک سکوی رایانش ابری[۵] برای اینترنت اشیا ارایه شده‌است. این معماری نوین، از ترکیب معماری بلاک‌چین و شبکه‌های نرم‌افزار محور برای بهبود عملکرد وایمنی اینترنت اشیا استفاده می‌کند. دراین کار، سعی شده‌است که مصرف انرژی و نسبت از دست رفتن بسته‌ها کاهش یافته و با استفاده از کنترلر شبکه‌های نرم‌افزارمحور به‌عنوان ایستگاه مرکزی، مسیریابی بهینه‌تری با استفاده از صف‌بندی جکسون با تاخیر کمتر و ایمنی بالاتر، برای انتقال داده‌ها به سمت ایستگاه مرکزی درنظرگرفته شود. بررسی پژوهش‌های پیشین نشان می‌دهد که در آنها تمرکز همزمان بر روی افزایش ایمنی در شبکه‌های اینترنت اشیا و کاهش مصرف انرژی به‌طور همزمان با درنظرگرفتن شبکه‌های نرم‌افزارمحور مورد توجه قرار نگرفته است. در این پژوهش، پرداختن به این دوموضوع به‌صورت همزمان، با استفاده از امکانات کنترلرها درشبکه‌های نرم‌افزارمحورو با استفاده از تکنولوژی‌های مدرن در هر دو زمینه از جمله نوآوری‌های مطرح است. برای رسیدن به این هدف، ابتدا گره‌های موجود در سیستم اینترنت اشیا خوشه‌بندی شده و سپس برای هر کدام از خوشه‌ها یک کنترلر از شبکه نرم‌افزارمحور به عنوان سرخوشه‌ای انتخاب می‌شود. این کنترلر، برای ساخت بلوک‌های اطلاعاتی از داده‌های ذخیره شده دستگاه‌های اینترنت اشیا درابر مورداستفاده قرارمی‌گیرد. با انتخاب درست مکان کنترلر درهرخوشه می‌توان علاوه بر کاهش مصرف انرژی، طول عمر شبکه را نیز افزایش داد. برای انتخاب کنترلرسرخوشه تکنیک‌های مختلفی ارایه شده‌است. انتخاب نادرست باعث افزایش مصرف انرژی و طول عمرشبکه می‌شود. دراین مقاله برای انتخاب بهترین مکان برای کنترلربرای بهینه‌سازی مصرف انرژی، از الگوریتم فراابتکاری بهینه‌سازی نهنگ[۶] استفاده شده و بهترین مکان برای قراردادن کنترلر شبکه نرم‌افزارمحور در خوشه‌ها تعیین می‌شود. درشبکه نرم‌افزارمحور، که از سه لایه زیرساخت، لایه کنترل و لایه برنامه‌های کاربردی تشکیل شده است، کنترل شبکه، از انتقال ترافیک مجزاشده و به‌طور مستقیم توسط کنترلر برنامه‌ریزی می‌شود. این شبکه‌ها، با استفاده از کنترلر وساختارجدید خود، راه را برای کاربردهای جدید و خلاقانه برای مدیران شبکه باز می‌کنند[۳]. با درنظرگرفتن هر کنترلر شبکه برای هر خوشه، این سرخوشه با یکپارچه‌کردن داده‌ها، آنها را به مقصد موردنظر ارسال می‌کند. درمتد پیشنهادی با خوشه‌بندی‌کردن و سپردن مسئولیت هرخوشه به کنترلرها، می‌توان در میزان مصرف انرژی صرفه‌جویی کرد. بانرمال کردن میزان انرژی اینترنت اشیا، شبکه پایدارتر شده و طول عمرآن طولانی می‌شود. دراین کار، امنیت داده‌های ذخیره‌شده درابر با رمزگذاری توسط الگوریتم دیفی‌هلمن[۷] وساخت بلوک‌های شبکه بلاک‌چین افزایش داده می‌شود[۴]. فناوری بلاک‌چین، یک پایگاه‌داده توزیع شده است که لیستی از رکوردهای اطلاعاتی با نام بلوک را درخود نگه می‌دارد. این پایگاه داده همانطور که در شکل ۲ نشان داده شده است، متشکل از زنجیره بلوک‌هایی است[۵].



شکل ۲: ساختار فرآیند درفن‌آوری بلاک‌چین

هربلوک حاوی یک هش رمزنگاری از بلوک قبلی، یک مهرزمانی و داده‌های تراکنش است. با تغییر اطلاعات در یک بلوک، مقدارهش آن تغییر می‌کند. از آنجا که در بلاک‌چین، بلوک‌ها بایکدیگر درارتباط هستند، این تغییر باعث عوض شدن مقدارهش بلوک‌های دیگر نیز می‌شود و بلوک‌ها نامعتبرمی‌شوند. ازاین‌رو با استفاده از هش می‌توان به‌درستی به اطلاعات یک بلوک پی‌برد[۶]. الگوریتم اجماع سهام[۸] یکی ازبخش‌های اصلی در تکنولوژی بلاک‌چین است و الگوریتم اجماع یکی از روش‌های رایج برای دستیابی به توافق در شبکه‌های بلاک‌چین به شمار می‌رود. این الگوریتم به‌عنوان جایگزینی برای الگوریتم اثبات کار[۹] طراحی شده تا مصرف انرژی را کاهش دهد و مقیاس‌پذیری شبکه را بهبود بخشد. در این الگوریتم، کاربران برای شرکت در اجماع، بخشی از دارایی‌های خود را قفل

می‌کنند و شانس انتخاب برای تولید بلوک جدید بر اساس مقدار و زمان قفل‌شدن دارایی‌ها است. افراد منتخب[10] بلوک جدید را ایجاد یا تأیید می‌کنند. در این الگوریتم، در صورت رفتار نادرست، بخشی یا کل دارایی قفل‌شده از دست می‌رود. این روش به دلیل مصرف انرژی کمتر و سرعت بالاتر نسبت به الگوریتم اثبات‌کارمحبوب‌تر است[۷]. دراین‌بخش چکیده گرافیکی مربوط به این مقاله درشکل ۳ دیده می‌شود.



شکل ۳: چکیده گرافیکی مقاله

دراین شکل ارتباط تکنولوژی‌های مورد استفاده در این مقاله که شامل محاسبات ابری، بلاک‌چین، شبکه‌های نرم‌افزارمحور و شبکه‌های اینترنت‌اشیا می‌باشد، درکناریکدیگر دیده می‌شود.

جزییات مربوط به این معماری پیشنهادی دربخش ۳ مطرح می‌شود. درادامه، ساختار مقاله به شرح زیر سازماندهی شده است: دربخش ۲ پژوهش‌های مرتبط در رابطه با بهینه‌سازی انرژی حسگرها در سیستم اینترنت اشیا و افزایش ایمنی با استفاده از شبکه‌های نرم‌افزار محور وبلاک‌چین بحث شده است. دربخش ۳، معماری پیشنهادی ارایه وجزئیات آن بررسی شده است. دربخش ۴، راه اندازی معماری پیشنهادی وپیاده‌سازی آن مطرح شده است. دربخش ۵، به تحلیل و ارزیابی نتایج ومقایسه آن با سایر الگوریتم‌های بهینه‌سازی مانند الگوریتم کلونی مورچگان[11]، الگوریتم ژنتیک[12] و پروتکل‌های مسیریابی دیگرمانند RPL[13] و CARP[14] پرداخته شده است. نهایتا، دربخش ۶، نتیجه‌گیری روش پیشنهادی مطرح و پیشنهادهایی برای کارهای آینده ارایه شده است.

۲- کارهای مرتبط

در این‌بخش ابتدا پژوهش‌های مرتبط با اینترنت‌اشیا و تأثیر معماری‌های نوین مانند بلاک‌چین و رایانش ابری بر ایمنی آنها مطرح شده است، همچنین تأثیر کنترلر شبکه‌های نرم‌افزارمحور در مدیریت کارآمد شبکه‌های اینترنت‌اشیا بررسی و پژوهش‌های مرتبط با کاهش مصرف انرژی مورد بحث و تحلیل قرار گرفته شده‌است. درپژوهش[۸]، آسایتامی وهمکارانش، یک معماری شبکه‌سازی مبتنی بر بلاک‌چین یکپارچه با شبکه نرم‌افزارمحور را برای شبکه‌های اینترنت‌اشیا صنعتی پیشنهاد داده‌اند. دراین معماری، چارچوبی برای پیاده‌سازی بلاک‌چین غیرمتمرکز یکپارچه با شبکه نرم‌افزارمحور برای کاربردهای‌صنعتی، به‌منظور دستیابی به مصرف بهینه انرژی و مبتنی بر انتخاب خوشه‌ها ارایه شده است. در راه‌حل پیشنهادی، دفترکل توزیع‌شده مبتنی بر بلاک‌چین، ثبات داده‌ها را درسراسر شبکه کنترل‌کننده تضمین می‌کند و یک رکورد از گره‌های اجرا شده در کنترل‌کننده نگه می‌دارد. برای ارزیابی روش پیشنهادی، به بررسی کارایی مدل پیشنهادی در میزان مصرف انرژی، تاخیر انتهابه‌انتها، توان عملیاتی ونتایج مربوط به هرکدام پرداخته شده است. لطیف وهمکاران در [۹]، یک پروتکل مسیریابی جدید برای شبکه‌های اینترنت اشیا با استفاده از معماری مبتنی بر بلاک‌چین پیشنهاد داده‌اند. دراین کاراز مزایای بالقوه یک سیستم بلاک‌چین وشبکه‌های نرم‌افزارمحور بهره برده شده است. درارزیابی این‌کار، عملکرد پروتکل مسیریابی پیشنهادی، نسبت به بردار فاصله تقاضا و مقصد، بردار فاصله ترتیبی و شبکه حسگر تلفن همراه امن مقایسه وارزیابی شده است. موتانا وهمکاران در[۱۰]، چارچوبی برای ساختار سیستم اینترنت اشیا ارایه داده‌اند که ازلایه کنترلی ومدیریت‌شده گره‌های مه توسط شبکه نرم‌افزارمحور، همراه با فن‌آوری بلاک‌چین استفاده می‌کند. سیستم پیشنهادی از شبکه نرم‌افزارمحور با کنترل‌کننده‌های توزیع شده و سوئیچ‌های اپن‌فلو توزیع‌شده استفاده می‌کند. دراین‌کار، یک مدل ترافیکی برای مدل‌سازی و تحلیل ترافیک بین بخش‌های مختلف شبکه، پیشنهاد و ارزیابی‌هایی براساس پارامترهای تاخیر، امنیت و استفاده از منابع درسیستم پیشنهادی انجام شده است. در[۱۱]، تکین وهمکاران، یک معماری جدید برای یک ساختمان هوشمند، شامل یک سیستم کنترل و رویکردهای خودکار، پیشنهادداده‌اند. دراین معماری، رویکردهای یادگیری ماشین مبتنی بر محاسبات ابری، مبتنی بر

محاسبات لبه و مبتنی بر دستگاه‌های اینترنت اشیا مقایسه شده‌اند. برای استنتاج نیز، رویکردی، برای اجرای الگوریتم‌های یادگیری ماشین بر روی دستگاه‌های کوچک اینترنت اشیا مانند واحدهای کنترل کننده میکرو انجام شد. ارزیابی عملکرد مقایسه‌ای نشان می‌دهد که استقرار الگوریتم درخت تصمیم بر روی دستگاه، نتایج بهتری از نظر زمان آموزش، زمان استنتاج و مصرف انرژی می‌دهد. در[۱۲]، رحمان و همکارانش، یک سکوی امن برمبنای ترکیب شبکه‌های نرم‌افزارمحور و اینترنت اشیا توزیع‌شده و مبتنی بر فناوری بلاک‌چین پیشنهاد داده‌اند. این پلتفرم مبتنی بر بلاک‌چین، از مزایای ترکیبی فن‌آوری‌های پیشرو استفاده می‌کند. دراین کار، یک الگوریتم بهینه برای انتخاب سرخوشه در میان خوشه‌ها ارائه شده است که به‌طور موثر در انرژی صرفه‌جویی می‌کند. دراین پژوهش، برای ارزیابی روش پیشنهادی وعملکرد معماری ارایه‌شده، از پارامترهای مختلف توان‌عملیاتی، نرخ ورود بسته، و زمان‌پاسخ استفاده شده‌است. در مقاله [۱۳]، ابوالهدی وهمکاران، چارچوب جدیدی مبتنی بر بلاک‌چین برای کاهش حملات سایبری پیشنهاد داده‌اند. این چارچوب از مفهوم قراردادهای هوشمند اتریوم برای تسهیل همکاری میان دامنه‌های مبتنی برشبکه‌های نرم‌افزارمحور و انتقال اطلاعات، به روشی غیرمتمرکز استفاده می‌کند. پیاده‌سازی معماری پیشنهادی در شبکه آزمایشی رسمی اتریوم به‌کارگرفته شده است و به بررسی انعطاف‌پذیری، کارایی و امنیت معماری پیشنهادی برای کاهش حملات در مقیاس بزرگ می‌پردازد. درمقاله [۱۴]، میسارا و همکاران، برای کاهش حملات مخرب به کنترلرها، یک معماری در یک محیط اینترنت‌اشیا با قابلیت مه و با استفاده از یک شبکه بلاک‌چین خصوصی پیشنهاد کرده‌اند. دراین معماری، از ماهیت غیرمتمرکز بلاک‌چین برای فعال‌کردن کنترل‌کننده‌های نرم‌افزارمحور محدود به منابع در شبکه استفاده‌شده و در صورتی که استخراج کننده‌ها قوانین جریان معیوب را شناسایی کنند، بلاک‌چین به دستگاه‌ها وگره‌های مه اجازه می‌دهد تا به قانون جریان قبلی بازگردند. درمعماری پیشنهادی، پارامترهای میزان مصرف پهنای باند وسرعت درمیان دستگاه‌ها بررسی و با استفاده ازنتایج بدست آمده، به‌بررسی تاخیر درسیستم و کاهش مصرف انرژی پرداخته شده است. درادامه، مدل پیشنهادی با روش‌های موجود مطرح‌شده در پژوهش‌های قبلی، از زوایای مختلف میزان مصرف انرژی گره‌ها، توزیع‌شدگی مدل، میزان ایمنی ایجاد شده از فناوری بلاک‌چین و میزان اعتمادپذیری‌کنترلردرجدول۱ مورد بررسی ومقایسه قرارگرفته شده است.

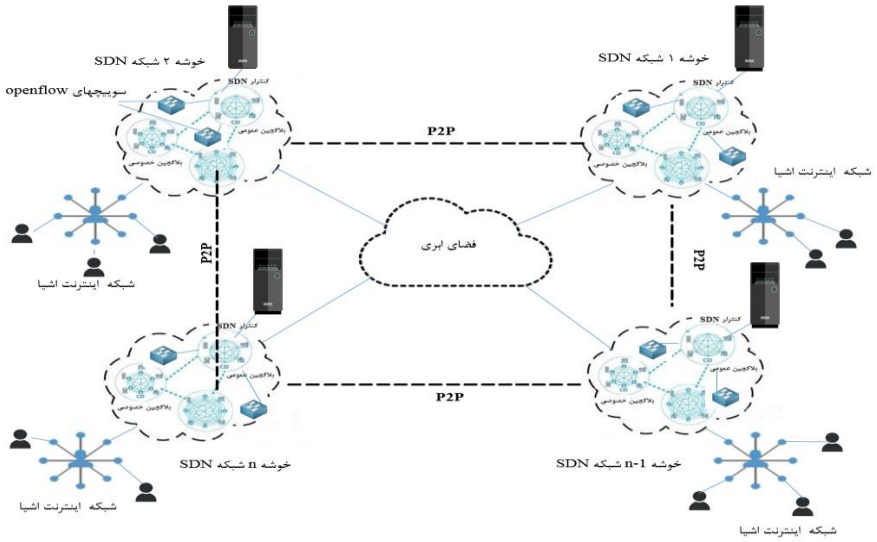
همانطور که دراین جدول دیده می‌شود، در کارهای تحقیقاتی ذکرشده برخی از پارامترهای مورد ارزیابی وجود ندارند، درحالیکه در معماری پیشنهادی تلاش شده است تا ازدیدگاه‌های بهینه‌سازی انرژی گره‌ها، توزیع شدگی وایمنی دستگاه‌های ثبت شده به‌صورت همزمان، مورد بررسی وتحلیل قراربگیرد.

جدول ۱: مقایسه پژوهش‌های انجام شده

منبع	بهینه‌سازی انرژی گره‌ها	توزیع‌شدگی	ایمنی دستگاه‌های ثبت‌شده
[۸]	✓	✓	×
[۹]	✓	✓	×
[۱۰]	✓	×	✓
[۱۱]	✓	✓	✓
[۱۲]	×	✓	✓
[۱۳]	✓	×	×
[۱۴]	×	×	✓
معماری پیشنهادی	✓	✓	✓

۳- معماری پیشنهادی

دراین مقاله برای افزایش ایمنی وکاهش مصرف انرژی در شبکه‌های اینترنت اشیا توزیع‌شده، از معماری ترکیبی بلاک‌چین وشبکه‌های نرم‌افزارمحور با نام BlockC-SDOT استفاده شده است. همانطورکه درشکل ۴ مشاهده می‌شود، این معماری، شامل نودهای شبکه اینترنت‌اشیا، کنترلرهای شبکه نرم‌افزارمحور، کنترلرهای مبتنی بر بلاک‌چین و سرورابری است. دراین معماری با استفاده از خوشه‌بندی شبکه و مدیریت کنترل هر خوشه توسط شبکه‌های نرم‌افزار محور، میزان مصرف انرژی و تاخیر در شبکه بهبود یافته و امنیت اطلاعات نیز تضمین می‌شود. برای بهینه‌سازی معماری، الگوریتم فراابتکاری نهنگ و الگوریتم صف‌بندی جکسون مورد استفاده قرارگرفته‌اند. دراین کار، از الگوریتم دیفی هلمن نیز برای رمزگذاری بسته‌های اطلاعاتی و افزایش ایمنی استفاده شده است. خوشه‌بندی با درنظرگرفتن کنترلر شبکه نرم‌افزارمحور به‌عنوان سرخوشه و انتخاب بهترین مکان برای قراردادن آن، از جمله نوآوری‌های این مقاله است که با استفاده از مزایای شبکه‌های نرم‌افزار محور و قدرت کنترلر دراین شبکه‌ها به‌کارگرفته شده است. درهرخوشه تعداد زیادی گره با یک کنترلر وجوددارد که هرگره اطلاعات خود را به این کنترلر منتقل می‌کند. انتخاب بهترین مکان براساس پارامترهایی مانند میزان انرژی هرحسگر وهزینه انتقال پکت‌ها انجام شده وبرای بهینه‌سازی مصرف انرژی از الگوریتم بهینه‌سازی فراابتکاری نهنگ استفاده می‌شود.

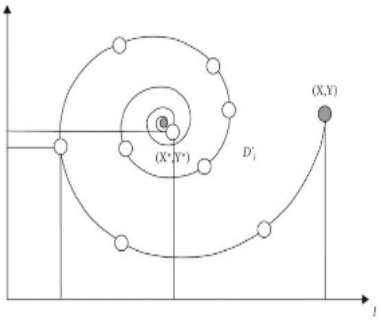


شکل ۴: معماری پیشنهادی BlockC-SDOT

برای این کار بهترین مکان برای قراردادن کنترلر شبکه نرم‌افزارمحور در خوشه‌ها تعیین می‌شود و با استفاده از کنترلر شبکه‌های نرم‌افزارمحور به‌عنوان ایستگاه مرکزی برای شبکه‌های اینترنت اشیا، مسیریابی بهینه‌تری با استفاده از صف‌بندی جکسون با تاخیر کمتر و ایمنی بالاتری برای انتقال داده‌ها به سمت ایستگاه مرکزی وجلوگیری از حملات سایبری توسط مهاجمین، درنظرگرفته می‌شود. دراین متد، کنترلرها، رهبری هرخوشه را به عهده می‌گیرند و ارتباط بین خوشه‌ها با استفاده از کنترلر هرخوشه برقرارمی‌شود وهرکدام به‌صورت یک خوشه جدید به بلاک‌چین عمومی اضافه می‌شوند. معیارهای موردنظر برای انتخاب مکان بهینه برای کنترلر، با هدف کاهش میزان مصرف انرژی گره‌ها در شبکه ومیزان تاخیر ارتباطی بین گره‌ها است. در این معماری، هر دستگاه اینترنت‌اشیا ابتدا در کنترلر شبکه نرم‌افزارمحور ثبت نام کرده و یک آدرس آی‌پی اختصاصی دریافت می‌کند. هر دستگاه دارای یک کلید عمومی و خصوصی است که برای انجام تراکنش‌های امن بلاک‌چین و اعمال سیاست‌های کنترل‌کننده استفاده می‌شود. هنگام ارسال داده‌ها به خوشه، دستگاه از کلید عمومی خود برای پخش و از کلید خصوصی برای تایید استفاده می‌کند. تراکنش‌ها در بلاک‌چین خصوصی ثبت و برای ذخیره‌سازی از فضای ابری استفاده می‌شود، سپس به بلاک‌چین عمومی افزوده می‌شود و ارتباط نظیر به نظیر با دیگر خوشه‌ها برقرار می‌شود. به‌منظور بهینه‌سازی مصرف انرژی، الگوریتم بهینه‌سازی نهنگ برای تعیین مکان قرارگیری کنترلر هر خوشه به‌کار رفته است.

۳-۱- بهینه‌سازی میزان مصرف انرژی با استفاده از الگوریتم بهینه‌سازی نهنگ

الگوریتم بهینه‌سازی نهنگ یک الگوریتم فراابتکاری مستقل از مسئله برنامه‌ریزی است که بر اساس رفتار اجتماعی نهنگ‌ها الهام گرفته شده است[۱۵]. شکل ۵، از استراتژی شکارشبکه حباب که توسط نهنگ‌های گوژپشت انجام می‌شود، برگرفته شده است و هر نهنگ به‌عنوان یک نقطه در فضای جستجو وفضای پارامترها در نظر گرفته می‌شود.



شکل ۵: رفتارتغذیه ای الگوریتم نهنگ

مراحل الگوریتم بهینه‌سازی نهنگ به‌صورت خلاصه به شرح زیر است:

۳-۱-۱- مقداردهی اولیه

درابتدا، موقعیت اولیه نهنگ‌ها به‌صورت تصادفی در فضای جستجو مشخص می‌شود وسپس ویژگی‌های هر نهنگ مثل موقعیت و سرعت آن در فضای جستجو تعیین می‌شوند.

۳-۱-۲- تعیین تابع هدف

تابع هدف برای ارزیابی کیفیت هر نهنگ در موقعیت فعلی محاسبه می‌شود. دراین مقاله، برای مشخص کردن تابع هدف مسئله خوشه‌بندی دستگاه‌ها در اینترنت اشیا، فرض شده است که منابع در دستگاه‌های مختلف در اینترنت اشیا به‌صورت یک ماتریس X با ابعاد $m \times n$ می‌باشند که m تعداد دستگاه‌ها و n تعداد ویژگی‌ها است. دراین تابع، نقش خوشه‌بندی دستگاه‌ها با توجه به موقعیت آن‌ها و مراکز خوشه‌ها توسط ماتریس C مشخص شده‌اند و تابع هدف به‌صورت رابطه ۱ تعریف می‌شود:

(۱)

$$f(X,C,E) = \sum_{i=1}^{m}\sum_{j=1}^{n} \frac{X_{ij}^2}{1+\sum_{k=1}^{n}(\frac{X_{ik}-C_{jk}}{\varepsilon})^2} + \gamma \sum_{i=1}^{m} E_i$$

دراین رابطه، X_{ij}، نشان دهنده ویژگی jام از دستگاه iام است. C یک ماتریس خوشه‌بندی با ابعاد $n \times k$ است که n تعداد گره‌ها و k تعداد خوشه‌ها است، ε نیز یک پارامترکنترل‌کننده است. دراین رابطه ،γیک پارامتر وزن‌دهی برای تعادل میان مصرف انرژی و کیفیت خوشه‌بندی بوده وE، نیز بردارمصرف انرژی در دستگاه‌ها است. تابع هدف ارائه‌شده برای بهینه‌سازی خوشه‌بندی دستگاه‌ها در اینترنت اشیا، به کمینه‌سازی مصرف انرژی و بهبود توزیع دستگاه‌ها در خوشه‌ها برای کاهش میزان تاخیر کمک می‌کند. این تابع هدف با هدف ایجاد خوشه‌هایی طراحی شده است که دستگاه‌های مشابه در آن‌ها قرار گیرند. یکی از وظایف کنترلر شبکه نرم‌افزارمحور در این معماری، اندازه‌گیری و بررسی میزان انرژی مصرفی گره‌ها در شبکه است. بسته به میزان انرژی مصرفی، دستگاه‌ها بسته‌ها را ارسال یا دریافت می‌کنند. مصرف انرژی در حسگرهای اینترنت اشیا تنها به پردازش داده‌ها محدود نمی‌شود، بلکه جمع‌آوری داده‌ها، پردازش اولیه، انتقال داده‌ها با کمترین میزان تاخیر، دریافت دستورات و تعامل با کاربر نیز بر آن تأثیر می‌گذارد.

ازآنجا که مصرف انرژی و تأخیر شبکه به عنوان دو هدف کلیدی دراین مقاله مورد توجه قرار گرفته‌اند، با استفاده از روابط ارائه‌شده، تابع هدف دوگانه‌ای طبق رابطه ۲ تعریف می‌شود که شامل کاهش مصرف انرژی و کمینه‌سازی تأخیر باشد. این تابع می‌تواند به صورت خطی یا غیرخطی تعریف شود و باید وزن‌دهی مناسبی به هر هدف داده شود تا تعادل بین این دو پارامتر ایجاد گردد.

$$T_{total}.\omega_1 + E_{total}.\omega_2 = ObjectiveFunction \qquad (۲)$$

دراین رابطه، ω_1 و ω_2 ضرایب وزن‌دهی هستند که باید براساس اهمیت نسبی هرهدف تنظیم شوند.E_{total} میزان انرژی مصرفی کلی و T_{total} میزان تاخیر درکل شبکه می باشد که مقادیرمربوط به آنها درادامه محاسبه می شود. استفاده از الگوریتم‌های چندهدفه مانند پارتو اپتیمال قادرند مجموعه‌ای از راه‌حل‌های بهینه را تولید کنند که تعادلی بین اهداف مختلف برقرار کند و ObjectiveFunction نیز تابع هدف مطرح درروش پیشنهادی است که هدف ازاین مقاله بهینه سازی آن می باشد. درروش پیشنهادی، شرایط محیطی همچون دما و رطوبت نیز می‌توانند مصرف انرژی را تغییر دهند. رابطه ۳ برای اندازه‌گیری میزان مصرف انرژی درکل شبکه استفاده می‌شود.

(۳)

$$E_{total} = \sum E_{ctrl_i} + E_{dev_i} + E_{comm} + E_{proc} + E_{env} + E_{other}$$

در این رابطه، E_{total} ، میزان کل انرژی مصرفی شبکه، E_{ctrli}، میزان انرژی مصرفی کنترل‌کننده خوشه iام، E_{devi} میزان انرژی مصرفی دستگاه‌های خوشه iام؛ E_{comm} میزان انرژی مصرفی در ارتباطات (شامل ارسال و دریافت داده‌ها)، E_{proc} میزان انرژی مصرفی در پردازش داده‌ها، E_{env} میزان انرژی مصرفی ناشی از عوامل محیطی، E_{other}، میزان سایر عوامل موثر بر مصرف انرژی (مانند انرژی مصرفی در مدیریت انرژی، سنسورها) ازعوامل تاثیرگذاردرمیزان انرژی مصرفی می‌باشد. دراین مقاله ازالگوریتم‌های خوشه‌بندی معمولاً در محیط‌های اینترنت اشیا به‌منظور گروه‌بندی دستگاه‌ها با ویژگی‌های مشابه و همچنین حفظ ارتباطات دستگاه‌های همسایه استفاده شده‌است. دراین کار، تابع هدف، تعادلی بین کیفیت خوشه‌بندی، تنوع در موقعیت دستگاه‌ها و کمینه کردن مصرف انرژی فراهم می‌کند. توزیع بهینه دستگاه‌ها در خوشه‌ها و اختصاص مناسب منابع انرژی در این سناریو می‌تواند به بهبود کارایی شبکه‌های اینترنت اشیا وکاهش مصرف انرژی کمک کند.

۳-۱-۳-مرحله به‌روزرسانی موقعیت نهنگ‌ها

با استفاده از رفتار اجتماعی نهنگ‌ها، موقعیت آن‌ها به‌روزرسانی می‌شود. نهنگی که بهترین کیفیت را دارد، موقعیت آن به‌عنوان موقعیت بهینه تلقی می‌شود. دراین پژوهش، برای به‌روزرسانی موقعیت نهنگ‌ها به منظور بهینه‌سازی خوشه‌بندی دستگاه‌های اینترنت اشیا و بهینه‌سازی مصرف انرژی ازرابطه ۴ استفاده می‌شود. این رابطه برای به‌روزرسانی موقعیت نهنگ i به منظور کمینه کردن تابع هدف ارائه شده است.

$$X_i = X_{best} - A \times (\frac{E_i}{E_{total}} \times \frac{X_i + X_{best}}{2}) \qquad (۴)$$

دراین رابطه فرض کنید X_i، موقعیت فعلی نهنگ i و X_{best} بهترین موقعیت تاکنون در مسئله باشند. همچنین فرض کنید E_i انرژی مصرفی مرتبط با موقعیت X_i و E_{total} مجموع انرژی مصرفی در کل شبکه باشد. دراین رابطه، A، یک عامل تصادفی است که برای حفظ تنوع در جستجوی بهینه استفاده می‌شود. بخش $\frac{E_i}{E_{total}}$ نیز میزان تاثیر مصرف انرژی دستگاه i را در به‌روزرسانی موقعیت تعیین می‌کند. این رابطه با توجه به مصرف انرژی دستگاه‌ها، نهنگ‌ها را به سمت موقعیت‌هایی هدایت می‌کند که توازن بهینه‌ای بین کیفیت خوشه‌بندی و مصرف انرژی فراهم آورد. استفاده از بهترین موقعیت دراین رابطه برای مشخص کردن بهترین مکان کنترلر درخوشه موردبررسی، مورد استفاده قرارمی‌گیرد.

۳-۱-۴- عملیات ترکیب

درالگوریتم مورد نظر، عملیات ترکیبی از جمله حرکت‌های اکتشافی و به‌روزرسانی‌های ترکیبی برای بهبود جستجو انجام می‌شود. این کارها برای حفظ تنوع جمعیت و همچنین بهبود نقاط بهینه تعادلی برای یافتن بهترین مکان کنترلرها در هرخوشه مورد استفاده قرارمی گیرند.

۳-۱-۵- شرط اتمام

درالگوریتم مورد نظر پس از تعداد خاصی از دورها یا تکرارها، الگوریتم ادامه یافته و پس از رسیدن به آن، متوقف شود. الگوریتم ۱ به معرفی روش پیشنهادی واستفاده از الگوریتم نهنگ دراین پژوهش پرداخته است.

الگوریتم ۱: شبکه کد متد پیشنهادی

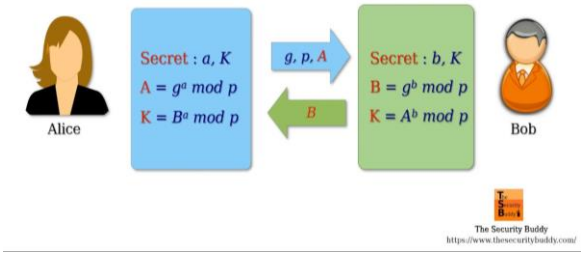
```
Input:
num_whales = 30, max_iter = 100, num_clusters = 5 , num_nodes = 100 , num_features = 5
# تولید شبکه اینترنت اشیا
iot_data = np. random. rand (num_nodes, num_features)
# تعیین کنترلرشبکه نرم‌افزارمحور به‌عنوان ایستگاه مرکزی درشبکه اینترنت اشیا
sdn_controller = np. mean (iot_data, axis=0)
# تعیین پارامترهای ساخت بلاکچین و ابر موردنظر
blockchain_key = " "کلید موردنیاز دربلاک‌چین
IaaS_cloud_credentials = {"username": "نام کاربری موردنظر",
"password": "پسورد موردنظر", "project_id": "شماره پروژه",
"auth_url": "auth_url"}
# موقعیت‌های تصادفی اولیه برای نهنگ‌ها در فضای جستجو
def initialize_whales(num_whales, num_clusters):
    return np. random. randint(0, num_clusters, size=num_whales)
# محاسبه تابع هدف برمبنای خوشه‌بندی
def calculate_fitness(whale, data, centroids):
f(X,C,E) = \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{X_{ij}^2}{1+\sum_{k=1}^{n}(\frac{X_{ik}-C_{jk}}{\varepsilon})^2} + \gamma\sum_{i=1}^{m}E_i
  labels = perform_sdn_clustering(data, centroids)
E_{total} = \sum E_{ctrl_i} + E_{dev_i} + E_{comm} + E_{proc} + E_{env} + E_{other}
  return f(data, labels,E_total)
# به‌روزرسانی موقعیت نهنگ‌هابر مبنای الگوریتم WOA
def update_position(whale, a, A, C, l, p_best, sdn_controller):
  r1 = np. random. random(num_clusters)
  r2 = np. random. random(num_clusters)
  D = np. abs(\frac{E_i}{E_{total}} * \frac{X_i + X_{best}}{2})
  X_i = X_best - A * D
  if np. random. random() < 0. 5:
    X_i += l * np. exp(a * r1) * np. cos(2 * np. pi * r2)
  return np. clip(X_i, 0, len(sdn_controller) - 1)
# تابع هدف اصلی برای الگوریتم بهینه‌سازی نهنگ
def whale_optimization_algorithm(data, num_clusters, max_iter, sdn_controller, a, A, C):
  whales = initialize_whales(num_whales, num_clusters)
X_best = np. zeros_like(sdn_controller)
  for k in range(num_clusters)])
      fitness[i] = calculate_fitness(whale, data)
      if fitness[i] > fitness[np. argmax(fitness)]:
        p_best = whale
# رمزنگاری با الگوریتم دیفی-هلمن در بلاکچین وذخیره در ابر
  encrypted_data = encrypt_data_with_diffie_hellman(data)
  stored_data = store_encrypted_data_in_iaas_cloud(encrypted_data)
# مسیریابی بهینه با استفاده از صف‌بندی جکسون
  routed_data = perform_optimal_routing(data, whales)
  for i in range(num_whales):
    l = 2 * np. random. random () – 1
    whales[i] = update_position(whales[i], a, A, C, l, p_best, sdn_controller)
  return p_best
# اجرای الگوریتم بهینه‌سازی روش پیشنهادی
optimal_clusters = whale_optimization_algorithm (iot_data, num_clusters, max_iter, sdn_controller, a=2, A=0. 5, C=1. 5)
#مشخص شدن خوشه‌های بهینه برای اینترنت اشیا
print("Optimal Clusters: ", optimal_clusters)
```

دربخش اول کد، پارامترهای ورودی الگوریتم پیشنهادی مانند تعداد نهنگ‌ها، خوشه‌ها، گره‌ها، حداکثر تکرار و ویژگی‌ها تعریف شده‌اند. همچنین، توابع مربوط به خوشه‌بندی شبکه و تعداد نودهای ذخیره‌سازی در شبکه تعیین شده‌اند. برای بهینه‌سازی مصرف انرژی

و خوشه‌بندی شبکه، از کنترلرهای نرم‌افزاری استفاده شده است. در این الگوریتم، برای رمزنگاری داده‌ها از الگوریتم دیفی‌هلمن و برای مسیریابی بهینه از صف‌بندی جکسون در ابر استفاده می‌شود. دربخش دوم، از الگوریتم بهینه‌سازی نهنگ برای تعیین مکان بهینه کنترلر در هر خوشه بهره‌برداری شده است.

۳-۲- ایمنی در معماری پیشنهادی

برای بالابردن ایمنی در معماری پیشنهادی، از الگوریتم رمزگذاری دیفی‌هلمن، برای بسته‌های منتقل‌شده درارتباطات استفاده می‌شود. این بسته‌ها پس از رمزگذاری توسط سوییچ‌های اپن فلو درشبکه نرم‌افزارمحور به سرور ابری زیرساخت منتقل‌شده ودراین فضای ابری ذخیره خواهند شد. الگوریتم دیفی‌هلمن، که درشکل ۶ نشان داده شده است، برای انتقال یک کلید مشترک بین فرستنده Alice و گیرنده Bob دریک کانال ناامن، نیازمند داشتن رضایت هردوطرف رابطه، است[۱۶].



شکل۶: پروتکل ارتباطی کلید درالگوریتم دیفی‌هلمن

دراین الگوریتم، فرستنده وگیرنده، از ریشه مشترک g و p که یک عدد اول دلخواه بوده، استفاده می‌کنند. هردوطرف، یک عدد دلخواه a وb را به‌صورت پنهانی درنظرگفته و با استفاده از عمل به‌توان رسانی پیمانه‌ای و مقادیر g وp، یک مقدار جدید A وB را محاسبه و برای طرف مقابل ارسال می‌کنند. سپس، برای رمزگذاری داده‌ها، هر دوطرف با مقادیر مشخص شده به محاسبه کلید مشترک K که در هردوطرف مشترک است، پرداخته و سپس اطلاعات رمز شده را در فضای ابری آپلود می‌کنند. پس از این عملیات، کنترلرهای مورد استفاده درهرخوشه شبکه، به مدیریت آنها می‌پردازند. دراین تکنیک، هر کنترلر برای تولید هش‌ها در بلوک تولید شده ازالگوریتم SHA-256 استفاده می‌کند [۱۷].

۳-۳- پروتکل مسیریابی بهینه پیشنهادی

پروتکل مسیریابی بهینه پیشنهادی دراین مقاله، براساس ساختار خوشه، محدودیت‌های انرژی و محاسباتی دستگاه‌های اینترنت اشیا پیشنهاد شده است. این پروتکل از ویژگی‌های امنیتی بلاک‌چین برای بهبود امنیت در راستای الزامات بهره‌وری انرژی و کنترل‌کننده برای احراز هویت و روش تایید در هر خوشه استفاده می‌کند. پروتکل پیشنهادی می‌تواند با تغییرات قابل‌توجهی که ممکن است در طول مسیریابی در هر خوشه از طریق کنترل‌کننده‌ها شبکه نرم‌افزارمحور موجود در همان خوشه یا کنترل‌کننده‌های مجاور در خوشه‌های همسایه اتفاق بیفتد، سازگارشود. دراین مقاله از مدل صف‌بندی جکسون طبق الگوریتم۲ برای پروتکل مسیریابی پیشنهادی استفاده شده است. شبکه صف‌بندی جکسون از N سیستم صف‌بندی مستقل M/M/1 تشکیل‌شده‌است و برای تخمین میانگین تاخیر انتهابه‌انتها و بررسی تغییرات تاخیر استفاده می‌شود[۱۸].

الگوریتم ۲: الگوریتم پروتکل مسیریابی

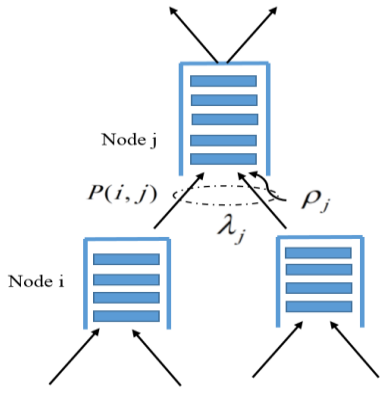
```
# پیاده‌سازی مدل صف‌بندی جکسون
class Queue-Jackson:
    def __init__(self):
        self. queue = []
    def enqueue(self, device_id):
        self. queue. append(device_id)
    def dequeue(self):
        if not self. is_empty():
            return self. queue. pop(0)
        else:
            return None
    def is_empty(self):
        return len(self. queue) == 0
# پیاده‌سازی کلاس کنترلرشبکه SDN
class Controller:
    def __init__(self):
        self. routing_table = {}
    def add_route(self, source_device,
destination_device, next_hop):
        key = (source_device, destination_device)
        self. routing_table[key] = next_hop
    def get_next_hop(self, source_device,
destination_device):
        key = (source_device, destination_device)
        return self. routing_table. get(key, None)
# پیاده‌سازی کلاس روتر در کنترلراینترنت اشیا
class Router-in-IOT:
    def __init__(self, device_id, queue_model,
sdn_controller):
        self. device_id = device_id
        self. queue_model = queue_model
        self. sdn_controller = sdn_controller
    def send_data(self, destination_device, data):
        next_hop= self. sdn_controller.
get_next_hop(self. device_id, destination_device)
# پیاده‌سازی انتقال داده
        if next_hop:
            print(f"Routing data from {self. device_id} to
{destination_device} via {next_hop}")
        else:
            print(f"No route found from {self. device_id} to
{destination_device}")
```

```
if __name__ == "__main__":
# ایجاد نمونه‌ای از کلاس صف‌بندی جکسون و کنترلر شبکه نرم‌افزارمحور
    queue_model = Queue-Jackson ()
  sdn_controller = Controller()

# پیکربندی و تنظیم کنترلرSDN با اطلاعات مسیریابی

  sdn_controller.  add_route("device1","device2",
"router1")
#ساخت روترهای شبکه اینترنت اشیا
router1 = IoTRouter("router1",queue_model,
sdn_controller)
# وارد کردن دستگاه‌های اینترنت اشیا درمدل صف‌بندی جکسون
  queue_model. enqueue("device1")
  queue_model. enqueue("device2")
# خارج کردن دستگاه از صف و ارسال داده
while not queue_model. is_empty():
    source_device = queue_model. dequeue()
    destination_device = queue_model. dequeue()
    data = " Data"
# ارسال داده از منبع به مقصد
    device = globals()[source_device]
    device. send_data(destination_device, data)
```

در این الگوریتم، مدل مسیریابی با استفاده از یک جدول مسیریابی در کنترلر شبکه نرم‌افزارمحور پیاده‌سازی شده است و ساختاری از کلاس‌ها برای مدل صف‌بندی جکسون، کنترلر شبکه و روترهای اینترنت اشیا ایجاد شده است. دراین روش با استفاده از کلاس Queue-Jackson، یک مدل از صف‌بندی جکسون طراحی شده است. از این مدل برای مدیریت دستگاه‌های موجود در صف استفاده می‌شود. دستگاه‌ها به‌ترتیب اولویت در این صف قرار می‌گیرند. سپس با استفاده از کنترلر شبکه نرم‌افزارمحور که مسئول مدیریت جدول مسیریابی است، اطلاعات مسیریابی از دستگاه مبدأ به دستگاه مقصد با استفاده از این کلاس، اضافه و دریافت می‌شوند. تحلیل وضعیت گره‌ها دراین صف‌بندی درشکل ۷ مورد بررسی قرارگرفته است:



شکل ۷: بررسی وضعیت گره‌ها درصف‌بندی جکسون

دررابطه‌های ۵ تا ۷، ارتباطات بین گره‌ها در سیستم صف‌بندی جکسون نشان داده‌شده‌است. دراین شبکه هنگامی که یک سرویس در گره i دریافت می‌شود، یک بسته با احتمال $p(i,j)$ به گره j می‌رود. این مقداراحتمال در رابطه (۵) محاسبه می‌شود[۱۹].

$$P(i,j) = \rho_1^i(1-\rho_1)\rho_2^j(1-\rho_2) \qquad (۵)$$
$$p_1 + p_2 = 1$$

دراین رابطه، ρ_i، نشان دهنده میزان ترافیک درگره i است که از رابطه (۶) بدست می‌آید.

$$\rho_1 = \frac{\lambda}{\mu_1 P_1}, \qquad \rho_2 = \frac{\lambda(1-P_1)}{\mu_2 P_1} \qquad (۶)$$

دررابطه (۷)، λ، مقدارمیانگین تولید بسته در گره‌های شبکه را نشان می‌دهد. برای محاسبه λ_j برای گره j از رابطه (۷)، استفاده می‌شود. ظرفیت صف درهرگره نامحدود درنظرگرفته‌می‌شود.

$$\lambda_1 = \frac{\lambda}{p_1} \quad , \quad \lambda_2 = \frac{\lambda(1-p_1)}{p_1} \qquad (۷)$$

دراین رابطه، p_i، میزان احتمال رفتن یک بسته از گره i با احتمال به گره j رانشان می‌دهد. دربخش اصلی این الگوریتم، یک نمونه از هر کلاس ایجاد شده و به‌هم متصل شده‌اند. سپس دستگاه‌ها به‌ترتیب اولویت به صف اضافه شده و داده‌ها از یک دستگاه به دستگاه دیگر ارسال می‌شوند. این کار با استفاده از اطلاعات مسیریابی از کنترلر شبکه نرم‌افزارمحور و مدل صف‌بندی جکسون انجام می‌شود. ازآنجا که اغلب دستگاه‌ها در اینترنت‌اشیا حساس به‌تاخیر هستند و تاخیرهایی در لینک‌های ارتباطی وجوددارد، رابطه ۸ میزان پارامتر تاخیر را اندازه‌گیری می‌کند.

$$T_{delay} = \sum_{i=1}^{N} T_i = \frac{2(\delta_i + 2\lambda_i + 1)}{(\delta_i - \lambda_i)(\delta_i + 3\lambda_i)}, \quad i \le L \qquad (۸)$$

دراین رابطه فرض می‌شود که N تعداد گره‌ها در صف‌بندی جکسون را نشان داده و L سطح هر گره درشبکه را نشان می‌دهد، دراین‌جا $\vec{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_N)$ بردار نرخ ورود میانگین خارجی بسته‌ها است که λ_j نرخ بسته‌های تولید شده توسط گره j و $\delta_i = (\delta_1, \delta_2, ..., \delta_n)$ بردار نشان دهنده نرخ خدمات است.

۳-۳-۱- تأخیرالگوریتم اجماع سهام

الگوریتم اجماع سهام به دلیل فرآیند انتخاب اعتبارسنج و زمان لازم برای اعتبارسنجی و تأیید بلاک، مقداری تأخیر اضافه می‌کند. این تأخیرشامل مجموعه‌ای از تاخیر اعتبارسنج طبق رابطه ۹، رابطه

تاخیر تولید وانتشاربلاک طبق رابطه ۱۰ و رابطه تاخیرتایید بلاک طبق رابطه ۱۱ می‌باشد[۲۰].

$$T_{selection=\frac{N}{f}} \quad (۹)$$

که دراین رابطه N تعداد گره‌ها و f میزان سرعت پردازش می‌باشد. همچنین رابطه تاخیرتولید وانتشار بلاک دربلاکچین طبق رابطه ۹ و ۱۰ بدست می‌آید.

$$T_{block} = T_{propagation} + T_{block-gen} \quad (۱۰)$$

$$T_{confirmation} = \frac{M}{\mu_{validation}} \quad (۱۱)$$

که دراین رابطه M گره‌های تاییدکننده و $\mu_{validation}$ نرخ تایید می‌باشد.

درمجموع این رابطه‌های تاخیر با یکدیگر، رابطه میزان تاخیر الگوریتم اجماع سهام را طبق رابطه ۱۲ بدست می‌آورند.

$$T_{PoS} = T_{selection} + T_{block} + T_{confirmation} \quad (۱۲)$$

درادامه مقدار تاخیر الگوریتم اجماع سهام در ترکیب با مقدارتاخیر در مدل صف‌بندی جکسون، مقدارتاخیر درکل سیستم وشبکه را طبق رابطه ۱۳ بدست می‌آورند.

$$T_{total} = T_{delay} + T_{PoS} \quad (۱۳)$$

۴- راه اندازی وارزیابی محیط

برای پیاده‌سازی معماری پیشنهادی، شبکه به مناطق مختلف تقسیم شده است و ویژگی‌ها وپارامترهای هر بخش در جدول ۲ مشخص شده است. در این کاربرد، حسگرهایی برای اندازه‌گیری پارامترهای محیطی مانند دما، رطوبت و آلودگی هوا در نقاط مختلف یک منطقه جغرافیایی نصب می‌شوند و داده‌های جمع‌آوری‌شده به یک پلتفرم مرکزی ارسال می‌شوند تا برای تحلیل و تصمیم‌گیری استفاده شوند. در هر منطقه، دستگاه‌های اینترنت‌اشیا، ابر، کنترل‌کننده شبکه نرم‌افزارمحور و بلاک‌چین‌های عمومی و خصوصی برای مدیریت و کنترل در نظر گرفته شده‌اند. توپولوژی معماری از ۳ دامنه اینترنت اشیا و یک کنترل‌کننده شبکه نرم‌افزارمحور برای هر دامنه استفاده می‌کند. ابتدا ۳۰ دستگاه اینترنت اشیا در یک خوشه با کنترلر شبکه نرم‌افزارمحور ارتباط برقرار می‌کنند و پس از ثبت‌نام، داده‌های رمز شده خود را به ابر ارسال می‌کنند. ظرفیت پردازشی ابر ۱۲۰۰ میلیون دستور بر ثانیه و ظرفیت ذخیره‌سازی آن ۲۴۰ گیگابایت است. این دستگاه‌ها به‌صورت بی‌سیم و با لینک اترنت ۱ گیگابیت‌برثانیه به یکدیگر متصل هستند. با افزایش تعداد نودها از ۳۰ تا ۱۸۰ نود، پارامترهای مختلف شبکه در سه ناحیه مختلف ارزیابی شده‌اند.

جدول ۲: پارامترهای شبیه‌سازی شبکه

	نام پارامتر	مقدار
پارامترهای عمومی	پکت آنالیزر	Wireshark
	پلتفرم ذخیره‌سازی ابر	OpenStack
پارامترهای شبکه نرم‌افزارمحور	شبیه ساز	Mininet
	نوع کنترلر	Ryu
	کنترلرهای شبکه نرم‌افزارمحور	۳
	سوییچ‌های اپن فلو	۳
	گیت وی	۲
	پروتکل مسیریابی	Openflow
پارامترهای بلاک‌چین	پلتفرم بلاک‌چین	Hyperledger Fabric
	مانیتورینگ شبکه	Prometheus
	هدر بلاک	۸۰ بایت
	ناحیه شبیه‌سازی	1000m *1000m
	تعداد دامنه‌های اینترنت اشیا	۳
	تعداد وسایل اینترنت اشیا	۳۰-۱۸۰
	سرعت وسایل اینترنت اشیا	۲۰m/s
	مدت زمان شبیه‌سازی	۵۰۰ ثانیه
	نرخ داده	۱۲ مگابایت برثانیه
	مقدارانرژی اولیه وسایل اینترنت اشیا	۱۲j-۱۵
	مقداراطمینان اولیه	۵j
	اندازه بسته انتقال گره	۵۱۲-۱۰۰ بایت

۵- تحلیل نتایج ومقایسه با پروتکل‌های مسیریابی

دربخش شبیه‌سازی، معماری پیشنهادی از نظر پارامترهایی مانند تأخیر، عملکرد CPU، توان عملیاتی، مصرف انرژی، نرخ ورود بسته‌ها و پهنای باند ارزیابی شده است. نتایج به‌دست‌آمده با پروتکل‌های مسیریابی RPL و CARP و همچنین الگوریتم‌های فرابتکاری ژنتیک و کلونی مورچگان مقایسه شده‌اند. پروتکل RPL برای سازگاری با شرایط شبکه و فراهم کردن مسیرهای جایگزین طراحی شده‌است[۲۱] و پروتکل CARP با استفاده از الگوریتم‌های حریصانه، انتقال بهینه بسته‌ها را انجام می‌دهد[۲۲]. الگوریتم‌های کلونی مورچگان [۲۳] و الگوریتم ژنتیک [۲۴]نیز برای بهینه‌سازی و بهبود راه‌حل‌ها در مسائل مختلف استفاده شده‌اند. نتایج بدست‌آمده، نشان‌دهنده برتری معماری پیشنهادی نسبت به سایر روش‌ها در شبکه اینترنت‌اشیا است.

۵-۱- مجموعه داده

مجموعه داده IOT-2020 مورد استفاده دراین مقاله شامل داده‌هایی است که ترکیبی از شبکه‌های اینترنت‌اشیا، شبکه‌های نرم‌افزارمحورو بلاک‌چین را شبیه‌سازی می‌کند. این مجموعه داده، به‌طور خاص برای شبیه‌سازی و ارزیابی شبکه‌های اینترنت اشیا در کنار شبکه‌های نرم‌افزارمحور ومعماری‌های بلاک‌چین طراحی شده است. داده‌های این مجموعه داده، شامل داده‌هایی از حسگرهای محیطی است که به شبکه اینترنت اشیا متصل بوده و شامل اطلاعاتی از دما، رطوبت، نور، فشار، و دیگر سنسورها می‌باشند. ویژگی‌های این مجموعه داده درجدول ۳ نشان داده شده‌اند[۲۵]. می‌توان از این داده‌ها برای بررسی عملکرد در شرایط محیطی مختلف مانند بیش از حد بارگذاری شبکه، انواع حملات، و تخصیص منابع در شبکه‌های اینترنت اشیا استفاده کرد.

جدول ۳: ویژگی‌های مجموعه داده IOT-2020

ویژگی	توضیحات
داده‌های ترافیکی اینترنت اشیا	شامل ویژگی‌های جریان ترافیک از دستگاه‌های مختلف اینترنت اشیا در محیط‌های مجهز به شبکه‌های نرم‌افزار محور است.
داده‌های تراکنش‌های بلاکچین	حاوی سوابق تراکنش‌های بلاکچین و فرآیندهای اعتبارسنجی است.
داده‌های برچسب‌گذاری‌شده از حملات	ارائه‌دهنده نمونه‌های برچسب‌گذاری‌شده از حملات منع سرویس توزیع شده و سایر تهدیدات امنیتی رایج در اینترنت اشیا است.
متریک‌های ارزیابی	شامل معیارهای دقت(Precision) ، بازیابی (Recall)، امتیاز F_1 و صحت (Accuracy) برای ارزیابی مدل‌های یادگیری ماشین است.
تشخیص حملات منع سرویس	داده‌های متمرکز برای تحلیل و کاهش حملات منع سرویس در محیط‌های اینترنت اشیا.
ایمنی دستگاه به دستگاه	ویژگی‌های مرتبط با ارتباط امن بین دستگاه‌های اینترنت اشیا با استفاده از بلاکچین
بخش بندی ترافیک	داده‌های بازرسی ترافیک از شبکه‌های نرم‌افزارمحور برای تحلیل عملکرد در شرایط ترافیک بخش‌بندی‌شده
مصرف انرژی	جزئیاتی درباره معیارهای مصرف انرژی در دستگاه‌های اینترنت اشیا با قابلیت بلاکچین و شبکه‌های نرم‌افزارمحور
داده‌های مربوط به تاخیر	اطلاعات مربوط به تاخیر شبکه در پیکربندی‌ها و بارهای مختلف
تشخیص آنومالی	داده‌های مفید برای شناسایی و کاهش ناهنجاری‌ها در ترافیک اینترنت اشیا

علاوه بر مجموعه داده IOT-2020 از مجموعه داده CICDDOS2019 نیز برای تشخیص حملات کامپیوتری منع‌سرویس توزیع‌شده دراینترنت‌اشیا دراین مقاله استفاده شده است.

CICDDoS2019 شامل داده‌های سالم و به‌روزترین حملات رایج DDoS است که شبیه به داده‌های واقعی در دنیای واقعی می‌باشد. این مجموعه داده همچنین نتایج تحلیل ترافیک شبکه با استفاده از CICFlowMeter-V3 را که دارای جریان‌های برچسب‌گذاری‌شده بر اساس زمان، آدرس‌های IP مبدأ و مقصد، پورت‌های مبدأ و مقصد، پروتکل‌ها و نوع حمله است، شامل می‌شود[۲۶].

۵-۲-پیاده‌سازی سیستم

دراین‌بخش سخت‌افزارها و نرم‌افزارهای مورد نیاز برای پیاده‌سازی چارچوب پیشنهادی مورد بررسی قرارمی‌گیرند. جدول ۴ مشخصات سیستمی که روش پیشنهادی در آن پیاده‌سازی شده است را نشان می‌دهد.

جدول ۴: ویژگی‌های سیستم استفاده شده برای شبیه‌سازی

سخت افزار/ نرم افزار	مشخصات
سیستم عامل	**Linux Ubunto**
نوع سیستم عامل	**۶۴ بیتی**
حافظه RAM	**16 GB**
سی پی یو	**Intel R Core™ i7-4790 CPU @ 3. 60GHz 3. 60 GHz**

دراین پیاده سازی، ابتدا شبکه بلاکچین مبتنی بر Hyperledger Fabric 2. 5 با استفاده از Docker و Docker Compose راه‌اندازی شده و گره‌های مربوطه برای هر سازمان و گره کنترلر برای مدیریت تراکنش‌ها پیکربندی می‌شوند. همچنین قراردادهای هوشمند به زبان جاوااسکریپت برای ثبت و تأیید داده‌های حسگرهای اینترنت اشیا توسعه یافتند. برای مدیریت شبکه از شبیه‌سازی شبکه نرم افزارمحور با Mininet و کنترلر RYU استفاده شده است. الگوریتم‌های بهینه‌سازی شامل الگوریتم بهینه سازی نهنگ والگوریتم صف بندی جکسون به زبان پایتون پیاده‌سازی و با الگوریتم های کلونی مورچگان والگوریتم ژنتیک مقایسه شدند. نتایج ارزیابی پارامترهایی نظیر تأخیر، مصرف انرژی، نرخ ارسال بسته‌ها و پهنای باند با ابزار Prometheus، نشان‌دهنده بهبود عملکرد و امنیت معماری پیشنهادی بود. پیاده سازی مطرح شده به این ترتیب است که درمرحله اول پیاده سازی، به نصب docker طبق دستورات زیر می‌پردازیم:

```
sudo apt update
sudo apt update sudo apt install docker docker-compose
```

سپس در مرحله بعد به نصب Hyperledger Fabric پرداخته می‌شود. این کار با استفاده از دستور زیر انجام می‌شود.

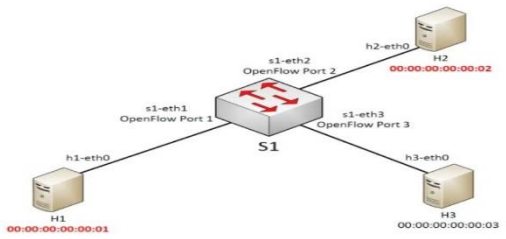
curl -sSL https: //bit. ly/2ysbOFE | bash -s

با استفاده از این اسکریپت، ابزارهای مورد نیاز برای Fabric دریافت می‌شود. این کانال‌ها در Hyperledger Fabric ، برای ارتباطات ایمن بین گره‌ها استفاده می‌شوند. این کانال‌ها برای برقراری ارتباط بین خوشه‌های اینترنت‌اشیا استفاده می‌شوند. درمرحله بعد برای هر خوشه یک سازمان تعریف می‌شود. هرسازمان شامل گره‌های مربوطه می‌باشد تا امنیت وقابلیت اعتماد افزایش یابد. پس از نوشتن قرارداد، قراداد در شبکه با استفاده از کد زیر اجرا می‌شود.

peer chaincode install -n iotcc -v 1. 0 -p. /chaincode peer chaincode instantiate -o orderer. example. com: 7050 -C mychannel -n iotcc -v 1. 0

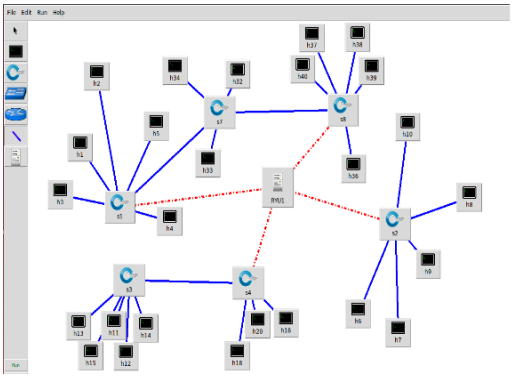
درادامه برای نوشتن واجرای ماژول‌های تشخیص حمله بر روی کنترلر RYU به زبان پایتون، نرم افزار Ecipse برروی کنترل نصب شده است. برای نوشتن این ماژول ابتدا وارد محیط ایکلیپس شده و با نصب پلاگین PyDev وتنظیمات انجام شده در بخش Package ، ماژول‌های مسیریابی و الگوریتم پیشنهادی، قابلیت اجرا در کنترلر RYU را پیداخواهند کرد. پس از این مرحله شبکه نرم افزارمحوربه بلاکچین متصل می‌شود. برای این کارگره‌های Fabric به کنترلر متصل می‌شوند واز REST API Fabric برای مدیریت داده‌ها در کنترلرشبکه نرم افزارمحور استفاده می‌شود. دراین شبکه با استفاده از شبیه‌ساز Mininet عملیات مدیریت وکنترل شبکه انجام می‌شود. درشکل ۸، مدیریت ورودی‌های جریان شبکه نرم افزارمحوررا روی Open vSwitch (OVS)، مشاهده می‌کنید که این کار با استفاده از فرمان ovs-ofctl انجام خواهدشد.

دراین شکل، ورودی‌های جریان روی یک سوییچ سازگار با OpenFlow رفتار بسته‌ها را کنترل می‌کنند. معمولاً این جریان‌ها به‌صورت پویا توسط یک کنترلر SDN نصب واجرا می‌شوند. شکل ۹، ارتباط هاست‌ها وسوییچ‌ها در شبکه رانشان می‌دهد. در این شکل، کنترلرها و میزبان‌ها به شبکه مورد نظر اضافه می‌شوند.

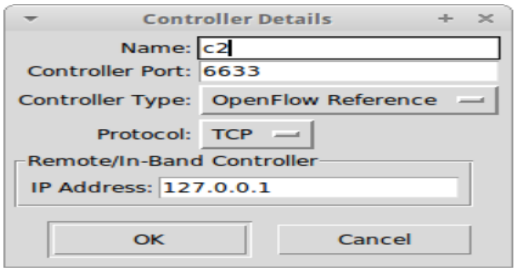


شکل ۸: ارتباط هاست‌های مختلف از طریق سوییچ در شبکه SDN

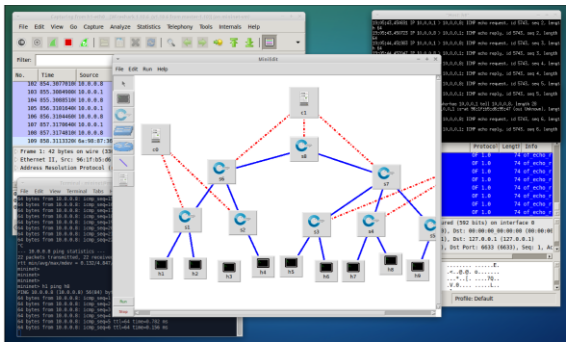
در این پیاده‌سازی، یک سوئیچ مجازی چندلایه در شبکه SDN استفاده شده است. در این شبیه‌ساز، پیکربندی وتنظیم اطلاعات هر کنترلر همان‌طور که در شکل ۱۰ نشان داده شده است، انجام گرفته وسپس ماژول‌های مورد نظر که درمقاله به صورت شبه‌کدهای الگوریتم‌های ۱ و ۲ می‌باشد برای آن نوشته واجرا می‌شود. پس از تنظیم واجرای ماژول‌های کنترلر، ابتدا پیکربندی پورت‌ها انجام شده و سپس اتصالات شبکه مطابق با الگوریتم پیشنهادی و به بهینه‌ترین شکل ممکن، همان‌طور که در نمودار شکل ۱۱ نشان داده شده است، برقرار می‌شوند.



شکل ۹: اضافه کردن هاست ها ، سوییچ ها و کنترلرها



شکل ۱۰: تنظیم کنترلرشبکه



شکل ۱۱: بهینه‌سازی شبکه

درحین پیاده سازی با استفاده از ابزارPrometheus مانیتورینگ عملکرد بلاکچین در شکل ۱۲ انجام شده وپارامترهایی مثل تأخیر، نرخ تراکنش، مصرف انرژی وسایرپارامترها اندازه‌گیری می‌شود. نتایج بدست آمده برای هرپارامتر دربخش تحلیل وارزیابی ذکرشده ونمودار مربوط به هرکدام از پارامترها رسم شده است.



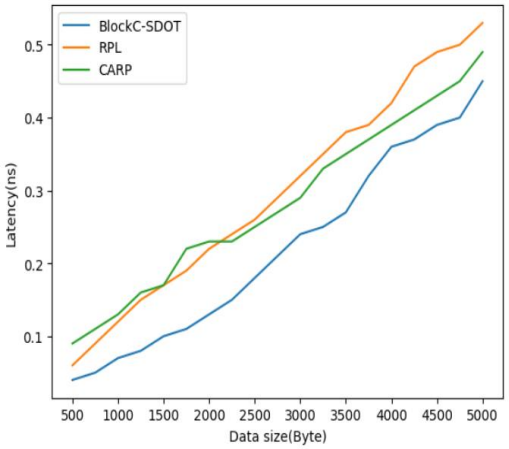
شکل ۱۲ : مانیتورینگ عملکرد بلاکچین با Prometheus

۵-۳- تحلیل وارزیابی پارامترهای موردارزیابی

دراین بخش به بررسی نتایج بدست‌آمده از شبیه‌سازی‌های انجام شده پرداخته شده و پارامترهایی چون تأخیر، توان عملیاتی، مصرف انرژی، پهنای‌باند، بررسی ایمنی وحملات سایبری، مورد ارزیابی قرار گرفته‌اند.درادامه این تحلیل‌ها مطرح شده است.

۵-۳-۱- تحلیل میزان‌تاخیر

در شبیه‌سازی روش پیشنهادی، پارامترهایی مانند نرخ ورودی بسته‌ها، زمان پردازش، طول صف، و زمان انتقال بین گره‌ها بررسی شده‌اند. داده‌ها از سرویس ابری و منابع اینترنت جمع‌آوری شده و شبیه‌سازی طی ۵۰۰ ثانیه در ۱۰ اجرای مجزا با تاخیر ۰/۰۰۰۱ ثانیه در لینک‌های ارتباطی انجام شده‌است. آزمایش‌ها شامل محاسبه تاخیر برای بسته‌هایی با اندازه ۱۰۰۰ تا ۵۰۰۰ بایت بوده و نتایج مقایسه‌ای نشان داده که معماری پیشنهادی در مقایسه با معماری‌های مبتنی بر پروتکل‌های RPL و CARP عملکرد بهتری دارد. دراین مرحله باید به این نکته توجه کرد که مدت زمان انتظار برای سرویس‌دهی به درخواست‌های موجود در شبکه باید مقدار بزرگتری از میانگین مدت زمان انتظاربرای درخواست داشته باشد. این موضوع باعث می‌شود که صف به‌صورت بی‌نهایت رشد نکند. شکل ۱۳، نتایج بدست‌آمده از میزان تاخیر را براساس اندازه بایت‌ها در شبکه برای معماری پیشنهادی و معماری‌های مبتنی بر مسیریابی RPL و CARP نشان داده است.



شکل ۱۳: مقایسه میزان تاخیر

همانطورکه در شکل دیده می‌شود، با افزایش زمان شبیه‌سازی به دلیل افزایش بسته‌های تولیدشده توسط هر منبع، میانگین تاخیر انتهابه‌انتها افزایش می‌یابد، هرچند در معماری جدید به دلیل وجود بلاک‌ها در بلاک‌چین و ارتباط آنها بایکدیگر و گستردگی آنها در همه خوشه‌های موجود در شبکه نسبت به پروتکل‌های مسیریابی مورد بررسی RPL وCARP ، با افزایش اندازه داده‌ها، میزان افزایش تاخیر کمتری در پاسخگویی سیستم حاصل‌شده‌است.

۵-۳-۲- ارزیابی حملات شبکه وبررسی امنیت شبکه

از آنجا که یکی از اهداف معماری پیشنهادی، افزایش ایمنی درشبکه‌های کامپیوتری است، دراین‌بخش به بررسی تشخیص حملات منع‌سرویس توزیع شده دربستر اینترنت‌اشیا بر روی معماری پیشنهادی و معماری بدون شبکه نرم افزارمحور پرداخته شده است. نتایج بدست آمده با استفاده از الگوریتمهای یادگیری ماشین مطرح مانند الگوریتم‌های BayesNet، J48، Logistic، Regression، RandomTree وReptree [۲۷] برروی مجموعه داده CICDDOS2019 بدست آمده است ومعیارهای TPR، FPR و میزان دقت ACC برای هرکدام ازحالت های درنظرگرفته شده به دست آمده است. این نتایج در جدول ۵ مشخص می‌باشند.

مقایسه نتایج نشان می‌دهد که روش پیشنهادی مبتنی بر شبکه نرم افزارمحور در تمامی معیارها، عملکرد بهتری نسبت به معماری بدون این شبکه دارد. به‌عنوان مثال، الگوریتم J48 در معماری همراه با شبکه نرم افزارمحوردقت ۸۹/۳۹و نرخ مثبت کاذب ۰/۶۲ داشته، در حالی که در معماری بدون شبکه نرم افزارمحور، دقت آن تنها ۴۰/۳۴و نرخ مثبت کاذب ۲/۱۲ بوده است. همچنین، الگوریتم RandomTree در روش همراه با شبکه نرم افزارمحور دقت بسیار

بالای ۹۹/۸۳ با نرخ مثبت کاذب ۰/۱۰ را در مقایسه با دقت ۷۱/۸۲ و نرخ مثبت کاذب ۲/۱۲ در معماری بدون شبکه نرم افزارمحور ارایه کرده است. به‌طور کلی، استفاده از شبکه نرم افزارمحور باعث بهبود قابل توجهی در شناسایی حملات شده و روش پیشنهادی مقاله را به گزینه‌ای بسیار مؤثرتر تبدیل کرده است.

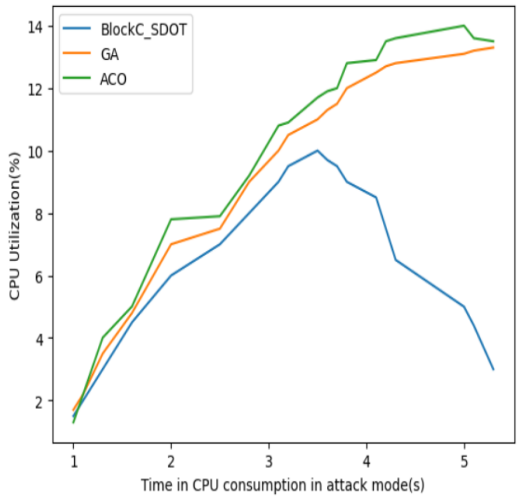
جدول ۵: مقایسه تشخیص حملات درروش های موردنظر

	الگوریتم	TPR	FPR	ACC
معماری بدون شبکه نرم افزارمحور	BayesNet	۸۶/۵۰	۱/۶۷	۸۶/۹۲
	J۴۸	۳۶/۴۳	۲/۱۲	۴۰/۳۴
	Logistic regression	۷۵/۰۴	۳/۴۷	۷۶/۰۲
	RandomTree	۶۹/۶۵	۲/۱۲	۷۱/۸۲
	RepTree	۸۳/۹۹	۳/۲۱	۸۵/۶۴
روش پیشنهادی	J۴۸	۹۷/۴۱	۰/۶۲	۸۹/۳۹
	BayesNet	۹۶/۹۵	۸/۲۶	۹۶/۱۵
	Logistic regression	۹۹/۰۸	۱۴/۴۹	۹۸/۵۷
	Naive Bayes	۹۶/۲۹	۶/۷۰	۹۴/۱۴
	RandomTree	۹۹/۶۴	۰/۱۰	۹۹/۸۳
	RepTree	۹۷/۳۵	۳/۹۶	۹۷/۹۷

۵-۳-۳- تحلیل عملکرد سی‌پی‌یو در محیط حملات مختلف

در این‌بخش عملکرد سی‌پی‌یو در زمان حملات منع سرویس توزیع‌شده با استفاده از مجموعه‌داده CICDDoS2019 تحلیل شده است. این مجموعه‌داده شامل اطلاعات ترافیک شبکه و انواع مختلف حملات کامپیوتری است و برای ارزیابی و بهبود سیستم‌های امنیتی استفاده می‌شود. در شکل ۱۴، میانگین مصرف سی‌پی‌یو در معماری‌های مختلف هنگام شروع حملات بررسی شده و نتایج الگوریتم نهنگ با الگوریتم‌های کلونی مورچگان و ژنتیک مقایسه شده است.

با بررسی نتایج مطرح شده در این شکل دیده می‌شود که تقریبا در ثانیه ۳، این حمله آغاز و درحال افزایش است. دراین بررسی سرعت حمله در طول زمان افزایش یافته است. با بررسی میزان استفاده از سی‌پی‌یو در طول زمان، مشاهده می‌شود که درمعماری پیشنهادی، این مقدار در یک دوره کوتاه افزایش یافته و سپس به‌سرعت به سطوح پایین‌تر سقوط می‌کند، درحالیکه در معماری‌های مبتنی بر الگوریتم ژنتیک والگوریتم کلونی مورچگان، به‌صورت افزایشی است. این موضوع نشان می‌دهد که" BlockC-SDOT " می‌تواند دفاع محکمی در برابر حملات منع سرویس توزیع‌شده، داشته باشد. درجدول۶ میزان درصد سی‌پی‌یو در معماری‌های مورد بررسی نشان داده‌شده‌است.



شکل ۱۴: بررسی وضعیت میزان استفاده از سی‌پی‌یو در زمان رخداد حمله

جدول ۶: درصد سی‌پی‌یو مصرفی

زمان (ثانیه)	BlockC-SDOT (%)	GA (%)	ACO (%)
۱	۱/۵	۱/۶	۱/۵
۲	۵/۸	۷/۵	۷/۸
۳	۸/۷	۹/۵	۱۰/۱
۴	۷/۵	۱۱/۹	۱۳/۳
۵	۲/۳	۱۳/۴	۱۴/۲

۵-۳-۴-تحلیل توان‌عملیاتی

برای بررسی متوسط نرخ دریافت پیام به‌صورت موفق در شبکه نیاز به‌بررسی وارزیابی پارامتر توان‌عملیاتی است. این پارامتر به بررسی تعداد عملیات‌هایی که در واحد زمان انجام‌شده، پرداخته و به نرخ‌داده‌ای اشاره‌دارد که یک گره می‌تواند بسته‌ها را بر روی شبکه ارسال‌کند. این پارامتر، در واقع، تعداد بسته‌هایی است که با موفقیت از مبدا به مقصد تحویل داده می‌شود. دررابطه ۱۴، توان‌عملیاتی براساس میزان نرخ‌بیت در شبکه مورد محاسبه، مطرح شده است[۲۸].

$$Throughput = \frac{\sum_{i=1}^{n} NBR_i}{t} \qquad (۱۴)$$

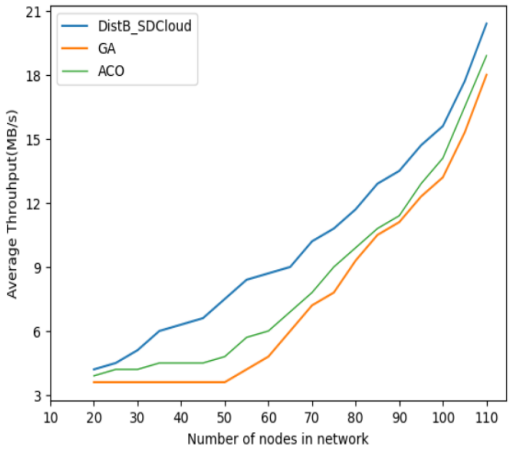
دراین رابطه NBR[۱۵] میزان نرخ‌بیت را در شبکه محاسبه می‌کند. درجدول ۷، میزان توان‌عملیاتی در معماری پیشنهادی ومعماری‌های مبتنی برالگوریتم‌های کلونی مورچگان و الگوریتم

ژنتیک برای بخشی از تعداد نودهای مختلف در شبکه و تعداد پکت‌های ارسالی نشان داده‌شده‌است.

جدول ۷: میزان توان‌عملیاتی

تعداد نودها در شبکه	تعداد پکت‌های ارسالی	BlockC-SDOT (kb/s)	GA (kb/s)	ACO (kb/s)
۱۵	۵۰	۵/۱	۳/۹	۴/۳
۳۰	۱۰۰	۵/۳	۳/۹	۴/۳
۴۵	۱۵۰	۶/۹	۳/۹	۴/۳
۶۰	۲۰۰	۸/۸	۵/۲	۵/۵
۷۵	۲۵۰	۱۱/۶	۸/۳	۹/۳
۹۰	۳۰۰	۱۴/۶	۱۱/۵	۱۱/۵
۱۰۵	۳۵۰	۱۷/۹	۱۶/۷	۱۷/۲

درشکل ۱۵، میزان توان عملیاتی برای بخشی ازتعداد نودهای مختلف در شبکه و تعداد پکت‌های ارسالی، نشان داده شده است. همانطور که در این شکل نشان داده شده است، با افزایش تعداد گره‌ها، توان‌عملیاتی افزایش‌یافته، درحالیکه وقتی تعداد گره‌ها کاهش می‌یابد، توان عملیاتی تقریبا ثابت باقی می‌ماند.



شکل ۱۵: مقایسه Throughput براساس تعداد گره‌ها در شبکه

همانطور که از شکل مشاهده می‌شود، میانگین مقادیر خروجی پروتکل پیشنهادی با الگوریتم نهنگ بیش از الگوریتم‌های ژنتیک و کلونی مورچگان افزایش‌یافته و ارزش آن با افزایش زمان از ۵ مگابایت به ۲۰ مگابایت افزایش می‌یابد. این موضوع می‌تواند عمدتا به دلیل دریافت مناسب بسته‌ها، افت بسته کم‌تر واحتمال خرابی لینک‌ها به‌خاطرافزایش اتصالات باشد.

۵-۳-۵- تحلیل میزان انرژی‌مصرفی

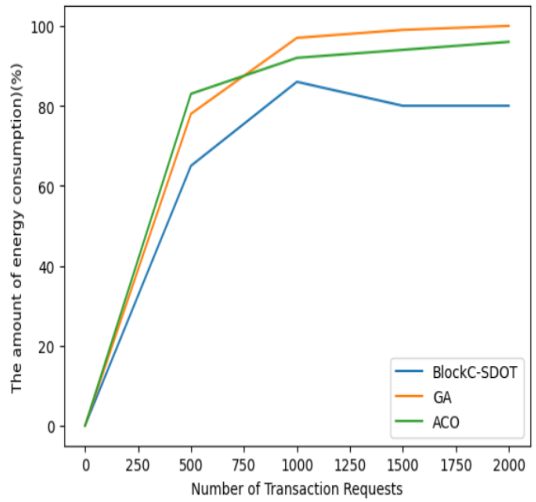
برای بررسی وتحلیل اثر مدل پیشنهادی بر بار کنترلرها وتاثیر آن در میزان مصرف انرژی در شبکه، آزمایش‌هایی انجام شده است.

درجدول ۸، براساس تعداد تراکنش‌های انجام شده در شبکه وتعداد نودهای هر خوشه، میزان مصرف انرژی در معماری روش پیشنهادی با معماری‌های دیگر مورد مقایسه قرارگرفته شده است. انرژی مورداستفاده برای رویکردهای مختلف در این جدول دیده شده‌است.

جدول ۸ : درصد انرژی‌مصرفی در معماری (٪)

تعداد درخواست‌های تراکنش	تعداد نودها	BlockC-SDOT	GA	ACO
۱۰۰	۳۰	۱۰	۱۴	۱۴
۵۰۰	۶۰	۷۷	۹۵	۹۳
۱۰۰۰	۹۰	۷۹	۹۶	۹۴
۱۵۰۰	۱۲۰	۷۸	۹۷	۹۲
۱۷۵۰	۱۵۰	۷۷	۹۸	۹۳
۲۰۰۰	۱۸۰	۷۶	۹۸	۹۴

دراین جدول برای تعداد گره‌های از ۳۰ تا ۱۸۰ میزان مصرف انرژی در معماری پیشنهادی با سایر معماری‌های مبتنی بر الگوریتم کلونی مورچگان والگوریتم ژنتیک مقایسه شده است. نتایج نشان از میزان مصرف انرژی کمتر در معماری پیشنهادی است. شکل ۱۶ عملکرد این مدل را نشان می‌دهد.



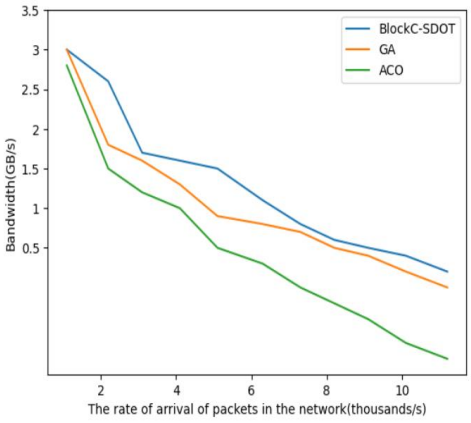
شکل ۱۶: مقایسه میزان مصرف‌انرژی براساس تعداد تراکنش‌های‌درخواستی در شبکه

نتایج بدست آمده نشان می‌دهد که مدل پیشنهادی با توزیع یکنواخت بار میان کنترلرها و بهینه‌سازی بار در تمامی شرایط، مصرف انرژی شبکه اینترنت اشیا را کاهش داده و عملکرد آن را بهبود می‌بخشد. کاهش بار کنترلرها منجر به تولید دمای کمتر توسط گره‌های حسگر و کاهش مصرف انرژی می‌شود. این مدل، با استفاده از رویکردی کارآمد، انرژی و به‌روزرسانی جداول مسیریابی بر اساس شرایط گره‌ها، عملکرد بهتری نسبت به الگوریتم‌های ژنتیک و کلونی مورچگان نشان داده است، زیرا این الگوریتم‌ها به

دلیل خرابی لینک و ارسال مجدد بسته‌ها مصرف انرژی بیشتری دارند. به طور کلی، پروتکل مسیریابی پیشنهادی در این پژوهش، کارایی بهتری در مقایسه با پروتکل‌های مشابه در شبکه اینترنت اشیا دارد.

۵-۳-۶- تحلیل نرخ ورود بسته‌ها وپهنای‌باند

دراین‌بخش، برای بررسی وضعیت ایمنی شبکه وحملات مربوطه، به بررسی میزان نرخ ورود بسته‌ها در شبکه و نحوه برخورد سیستم و وضعیت پهنای‌باند آن در معماری موردنظر پرداخته شده‌است. درشکل ۱۷، نتایج بدست آمده برای وضعیت پهنای باند، برای معماری پیشنهادی و معماری‌های مبتنی بر الگوریتم‌های کلونی مورچگان والگوریتم ژنتیک با میزان نرخ‌متفاوتی از ورود بسته‌ها را نشان می‌دهد.



شکل ۱۷: مقایسه پهنای‌باند

همانطورکه دراین شکل مطرح شده‌است، نتایج بدست‌آمده نشان می‌دهد که با افزایش نرخ ورود بسته‌ها در شبکه، میزان پهنای باند در شبکه کاهش می‌یابد. همچنین، درجدول ۹، نتایج بدست آمده برای وضعیت پهنای باند، برای هردو معماری با میزان نرخ متفاوتی، ورود بسته‌ها را نشان می‌دهد. نتایج بدست آمده توسط الگوریتم نهنگ با نتایج بدست آمده ازالگوریتم‌های کلونی مورچگان والگوریتم ژنتیک مقایسه شده است.

جدول ۹: پهنای باند (تعداد درخواست / ثانیه)

نرخ ورود بسته‌ها (thousands/s)	BlockC-SDOT	GA	ACO
۲	۲/۶	۲/۱	۱/۵
۴	۲/۳	۱/۳	۱/۱
۶	۱/۷	۰/۷	۰/۴
۸	۱/۶	۰/۵	۰/۲
۱۰	۱/۶	۰/۴	۰/۱

از آنجا که افزایش نرخ ورود بسته‌ها می‌تواند ناشی از وجود حمله در سیستم باشد، دراین معماری، پهنای‌باند با افزایش نرخ ورود بسته‌ها تغییر زیادی نداشته است، این در حالی است که پهنای باند درمعماری مبتنی بر الگوریتم‌های ژنتیک و الگوریتم کلونی مورچگان، به‌شدت کاهش یافته‌است. این نتایج نشان می‌دهدکه معماری پیشنهادی در مقابل حملات می‌تواند پایدار بماند و دچار تغییر زیادی در وضعیت پهنای‌باند نشود.

۵-۴- مقایسه معماری پیشنهادی با درنظرگرفتن کنترل شبکه نرم افزارمحور و بدون درنظرگرفتن شبکه‌های نرم افزارمحور

برای مقایسه عملکرد معماری پیشنهادی این مقاله، که ازترکیب شبکه‌های نرم‌افزار محور با تکنولوژی‌های دیگراستفاده می‌کند، با حالتی که از شبکه نرم افزارمحوراستفاده نشده است، تعدادی پارامتر اصلی برای هر دو حالت در نظر گرفته شده است. این پارامترها شامل میزان تاخیر، مصرف انرژی، عملکرد CPU، توان عملیاتی، نرخ ورود بسته‌ها و پهنای باند درهردوحالت می‌باشند. در ادامه در جدول ۱۰ مقایسه‌ای بین نتایج برای هر یک از این پارامترها در هر دو حالت مختلف شبکه مطرح شده است.

جدول ۱۰: مقایسه معماری‌ها با وبدون شبکه‌های نرم افزارمحور

پارامتر	معماری با استفاده از شبکه نرم افزارمحور	معماری بدون استفاده از شبکه نرم افزارمحور
میزان تاخیر(ms)	۳۰	۷۰
مصرف انرژی(mj)	۵۰	۱۲۰
عملکرد(CPU)	۸۵	۶۰
توان عملیاتی (Gbps)	۵	۳
نرخ ورود بسته ها (PPS)	۵۰۰۰۰۰	۳۵۰۰۰۰
پهنای باند(Mbps)	۱۰۰۰	۷۴۰

نتایج به دست آمده از جدول نشان می‌دهد که استفاده از شبکه‌های نرم‌افزارمحور موجب کاهش چشمگیر تأخیر از ۷۰ میلی‌ثانیه به ۳۰ میلی‌ثانیه می‌شود، که نشان‌دهنده بهینه‌سازی مسیرهای انتقال و مدیریت بهتر ترافیک است. همچنین، مصرف انرژی با استفاده از شبکه‌های نرم‌افزارمحور از ۱۲۰ میلی‌ژول به ۵۰ میلی‌ژول کاهش یافته، که بهینه‌سازی انرژی را تایید می‌کند. در مورد عملکرد CPU، استفاده از این شبکه‌ها باعث افزایش عملکرد از ۶۰ درصد به ۸۵ درصد شده است که نشان‌دهنده کاهش بار پردازشی و بهینه‌سازی منابع است. به‌طورکلی، نتایج نشان‌دهنده برتری شبکه‌های

نرم‌افزارمحور در بهبود کارایی شبکه، کاهش تأخیر، صرفه‌جویی در انرژی و افزایش توان عملیاتی است.

۵-۵- مقایسه دستاوردهای روش پیشنهادی دراین مقاله با مقالات پژوهشی مشابه

دراین‌بخش به مقایسه دستاوردهای حاصل از این پژوهش باسایر مقالات مشابه دراین حوزه پرداخته شده‌است. این مقایسه از جنبه های مختلف ازجمله نوع معماری، امنیت، میزان تاخیردرشبکه ، میزان مصرف انرژی وپروتکل‌های مورد توافق انجام شده است. نتایج بدست‌آمده در جدول ۱۱ مطرح شده‌اند.

جدول ۱۱: مقایسه دستاوردهای روش پیشنهادی با پژوهش‌های مشابه

جنبه مقایسه	مقاله موردنظر	مقاله [۲۹]	مقاله [۳۰]	مقاله [۳۱]
تمرکز اصلی	کاهش تاخیر، بهبودامنیت داده، کاهش مصرف انرژی درمعماری بلاکچین اینترنت اشیا محور	تاثیربلاکچین و شبکه های نرم افزارمحوربرحریم خصوصی، اینترنت اشیا	افزایش امنیت اینترنت اشیا ابری با تمرکز بر جلوگیری از حملات	بهبودامنیت و اعتماد در ارتباطات دستگاه به دستگاه در شبکه های نرم افزارمحور
نوع معماری پیشنهادی	ترکیبی از بلاکچین، شبکه نرم افزارمحور ورایانش ابری	ترکیب شبکه نرم افزارمحور و بلاکچین	ترکیب بلاکچین و شبکه نرم افزارمحوربرای مقابله با حملات	استفاده از بلاکچین برای تقویت ارتباطات D2D در شبکه نرم افزارمحور
امنیت	الگوریتم دیفی هلمن، امضاهای دیجیتال، مکانیزم BFT	کاهش نقاط شکست، بلاکچین برای امنیت	استفاده از بلاکچین برای جلوگیری از حملات	استفاده از بلاکچین و قراردادهای هوشمند برای احراز هویت دستگاه‌ها
تاخیر شبکه	تحلیل و کاهش تاخیر با استفاده از خوشه بندی و SDN	بدون تمرکز برتاخیر	بدون اشاره مشخص به تاخیر	بدون اشاره به کاهش تاخیر درارتباطات D2D
مصرف انرژی	کاهش مصرف انرژی حسگرها از طریق الگوریتم نهنگ و صف بندی جکسون	بدون اشاره به مصرف انرژی	بدون تمرکز برمصرف انرژی	بدون بررسی مصرف انرژی
مقایسه با روش‌های دیگر	مقایسه با ACO، RPL، GA، CARP	بدون مقایسه عملی	بدون مقایسه عملی	بدون مقایسه عملی
شبیه سازی و ارزیابی	شبیه سازی دقیق برای تحلیل تاخیر، مصرف انرژی، توان عملیاتی و سایرپارامترها	آزمایش محدود برای جلوگیری از حملات	بدون شبیه سازی	اثبات امنیتی محدود در ارتباطات
پروتکل‌های توافق	استفاده از الگوریتم اجماع اثبات سهام برای توافق گره ها	بدون تمرکز خاص برپروتکل‌های توافق	بررسی کلی پروتکل‌های توافق	استفاده از قراردادهای هوشمند برای توافق

درادامه مزیت های روش پیشنهادی نسبت به پارامترهای درنظرگرفته شده در جدول ۱۱ را با روش پیشنهادی دراین مقاله مورد نظر مقایسه کرده ودرجدول ۱۲ ، مزایای روش پیشنهادی را در پارامترهای موردنظرمطرح می کنیم.

جدول ۱۲: مزیت های روش پیشنهادی دربررسی پارامترها

پارامتر	مزایای روش پیشنهادی نسبت به ۳ مقاله موردنظر
تمرکز اصلی	روش پیشنهادی دراین مقاله، سه هدف اصلی امنیت، تاخیر و انرژی را به طور همزمان پوشش می‌دهد درحالی که مقالات دیگر به یک یا دوهدف متمرکز هستند
نوع معماری پیشنهادی	معماری پیشنهادی ترکیب جامع تری از فناوری‌ها ارایه داده است و شامل مکانیزم‌های بهینه سازی قویتری است
امنیت	مقاله پیشنهادی جزییات بیشتری در مورد الگوریتم‌های امنیتی و توافق‌های امنیتی ارایه می‌دهد
تاخیر شبکه	مقاله پیشنهادی کاهش تاخیر را به عنوان یکی از اهداف اصلی مطرح کرده و با استفاده از شبیه سازی تایید کرده است
مصرف انرژی	مقاله پیشنهادی بهینه سازی مصرف انرژی را به طورخاص بررسی کرده است
مقایسه با روش‌های دیگر	مقاله پیشنهادی از مقایسه‌های کمی با الگوریتم‌ها و پروتکل‌های مشابه استفاده کرده و کارایی را به طور دقیق نشان داده است
شبیه سازی و ارزیابی	مقاله پیشنهادی با شبیه سازی گسترده عملکرد معماری پیشنهادی را به طورجامع ارزیابی کرده است
پروتکل‌های توافق	مقله پیشنهادی جزییات بیشتری از پروتکل‌های توافق ارایه داده است

مقایسه میان مقالات دراین جدول، نشان می‌دهد که رویکرد ارائه‌شده در این تحقیق با ارائه معماری ترکیبی BlockC-SDOT که از ترکیب بلاکچین، شبکه‌های نرم‌افزارمحورو رایانش ابری بهره می‌برد، توانسته است به‌طور همزمان مسائل کلیدی اینترنت اشیا شامل امنیت، کاهش تأخیر، و بهینه‌سازی مصرف انرژی را هدف قرار

دهد. برخلاف سایر تحقیقات که به یکی از این جنبه‌ها تمرکز دارند، این پژوهش با به‌کارگیری روش‌های بهینه‌سازی نوینی همچون الگوریتم نهنگ و مکانیزم صف‌بندی جکسون و با استفاده از پروتکل اجماع پیشرفته‌ای مانند PoS، توانسته است بهبود قابل توجهی در عملکرد و کارایی شبکه ارائه کند.

۶- نتیجه‌گیری وپیشنهادهای آینده

در این مطالعه، معماری ترکیبی BlockC-SDOT که از ترکیب بلاک‌چین و شبکه‌های نرم‌افزارمحور برای بهبود عملکرد و امنیت اینترنت‌اشیا استفاده می‌کند، نتایج مثبتی را به همراه داشت. نتایج شبیه‌سازی‌ها نشان داد که این معماری در مقایسه با روش‌های موجود در کاهش تأخیر، بهبود عملکرد CPU در حملات مختلف، کاهش مصرف انرژی و افزایش امنیت، برتری دارد. همچنین، استفاده از الگوریتم‌های بهینه‌سازی همچون نهنگ و صف‌بندی جکسون، به بهینه‌سازی معماری کمک کرده و نتایج قابل قبولی را در تحلیل‌های مختلف به دست داد. با این حال، محدودیت‌هایی در این مطالعه وجود دارد. یکی از محدودیت‌ها، وابستگی به شبکه‌های نرم‌افزارمحور است که ممکن است در شرایط خاص باعث پیچیدگی در پیاده‌سازی و مدیریت شود. همچنین، محدودیت‌هایی در مقیاس‌پذیری این معماری در شبکه‌های بزرگ و پیچیده وجود دارد که نیاز به تحقیقات بیشتر دارد. علاوه بر این، در نظر گرفتن تعداد زیادی گره در شبیه‌سازی‌ها ممکن است باعث ایجاد مشکلات در زمان‌بندی و همگام‌سازی سیستم شود. همچنین، در این مطالعه، تنها از الگوریتم‌های خاص برای بهینه‌سازی استفاده شده و بررسی سایر الگوریتم‌ها برای ارزیابی کارایی معماری می‌تواند نتایج متفاوتی به همراه داشته باشد. برای تحقیقات آینده، پیشنهاد می‌شود که به بررسی قابلیت مقیاس‌پذیری معماری پیشنهادی در شبکه‌های بزرگ‌تر پرداخته شود و الگوریتم‌های جدیدی برای بهینه‌سازی بیشتر عملکرد و کاهش تأخیر در شرایط واقعی مورد آزمایش قرار گیرند. همچنین، بررسی نحوه تعامل بهتر میان بلاک‌چین و شبکه‌های نرم‌افزارمحور در محیط‌های پیچیده‌تر می‌تواند به بهبود عملکرد و امنیت این معماری کمک کند.

مراجع

[1] R. Chataut, A. Phoummalayvane, and R. Akl, “Unleashing the Power of IoT: A Comprehensive Review of IoT Applications and Future Prospects in Healthcare, Agriculture, Smart Homes, Smart Cities, and Industry 4. 0,” *Sensors*, vol. 23, no. 16, p. 7194, Jan. 2023, doi: https: //doi. org/10. 3390/s23167194.

[2] Swati Sandeep Chopade, Hari Prabhat Gupta, and T. Dutta, “Survey on Sensors and Smart Devices for IoT Enabled Intelligent Healthcare System,” Jun. 2023, doi: https: //doi. org/10. 1007/s11277-023-10528-8.

[3] N. Gupta, M. S. Maashi, S. Tanwar, S. Badotra, M. Aljebreen, and S. Bharany, “A Comparative Study of Software Defined Networking Controllers Using Mininet,” *Electronics*, vol. 11, no. 17, p. 2715, Jan. 2022, doi: https: //doi. org/10. 3390/electronics11172715.

[4] V. A. Yakovlev, D. Z. Satybaldina, E. Egamberdiyev, and Y. Seitkulov, “Authentication and key distribution protocol based on Diffie-Hellman algorithm and physically unclonable function,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 22, no. 2, pp. 290–301, Apr. 2024.

[5] H. Farran, D. Khoury, and L. Bokor, “A comprehensive survey on the application of blockchain/hash chain technologies in V2X communications,” INFOCOMMUNICATIONS JOURNAL, vol. 14, no. 1, pp. 24–35, 2022, Accessed: Sep. 13, 2023. [Online]. Available: http: //real. mtak. hu/144581/.

[6] J. Xu, C. Wang, and X. Jia, “A Survey of Blockchain Consensus Protocols,” ACM Computing Surveys, Jan. 2023, doi: https: //doi. org/10. 1145/3579845.

[7] Wang, N., Zhou, Z., Liu, J., Deng, L., & Fu, J. (2024). Secure and Distributed IoV Data Sharing Scheme based on a Hybrid PoS Blockchain Protocol. *IEEE Transactions on Vehicular Technology*.

[8] S. Asaithambi et al. , “An Energy-Efficient and Blockchain-Integrated Software Defined Network for the Industrial Internet of Things,” Sensors, vol. 22, no. 20, p. 7917, Oct. 2022.

[9] S. A. Latif et al. , “AI-empowered, blockchain and SDN integrated security architecture for IoT network of cyber physical systems,” Computer Communications, vol. 181, pp. 274–283, Jan. 2022, doi: https: //doi. org/10. 1016/j. comcom. 2021. 09. 029.

[10]Ammar Muthanna et al. , “Secure IoT Network Structure Based on Distributed Fog Computing, with SDN/Blockchain,” Jan. 2019, doi: https: //doi. org/10. 20944/preprints201901. 0305. v1.

[11] N. Tekin, A. Acar, A. Aris, S. Uluagac, and V. Cagri , “Energy consumption of on-device machine learning models for IoT intrusion detection.

[12]A. Rahman, M. Jahidul Islam, Z. Rahman, and M. Mahfuz , “DistB-Condo: Distributed Blockchain-Based IoT-SDN Model for Smart Condominium,” Nov. 02, 2020.

[13] Z. A. El Houda, A. Hafid, and L. Khoukhi, “Co-IoT: A Collaborative DDoS Mitigation Scheme in IoT Environment Based on Blockchain Using SDN,” IEEE Xplore, Dec. 01, 2019.

[14]S. Misra, P. Kumar Deb, N. Pathak, and A. Mukherjee, “Blockchain-Enabled SDN for Securing Fog-Based Resource-Constrained IoT,” Aug. 10, 2020.

[15] Seyed Mostafa Bozorgi, Mahdi Rohani Hajiabadi, A. Asghar, and Arun Kumar Sangaiah, “Clustering based on whale optimization algorithm for IoT over wireless nodes,” *Soft computing (Berlin. Print)*, vol. 25, no. 7, pp. 5663–5682, Jan. 2021.

[16]A. F. -X. Ametepe, A. S. R. M. Ahouandjinou, and E. C. Ezin, “Robust encryption method based on AES-CBC using elliptic curves Diffie–Hellman to secure data in wireless sensor networks,” Wireless Networks, vol. 28, no. 3, pp. 991–1001, Jan. 2022.

[17] M. D. Mohanty *et al.* , “Design of Smart and Secured Healthcare Service Using Deep Learning with Modified SHA-256 Algorithm,” *Healthcare*, vol. 10, no. 7, p. 1275, Jul. 2022.

[18]A. Ahmadi-Javid and M. Fathi, “Design of multi-service systems with facilities functioning as open Jackson queueing networks: application to online shopping stores,” *OR Spectrum*, Jun. 2022.

[19] W. Liang, L. Cui, and F. P. Tso, “Low-latency service function chain migration in edge-core networks based on open Jackson networks,” *Journal of Systems Architecture*, vol. 124, p. 102405, Mar. 2022, doi: https: //doi. org/10. 1016/j. sysarc. 2022. 102405.

[20] Wendl, M., Doan, M. H., & Sassen, R. (2023). The environmental impact of cryptocurrencies using proof of work and proof of stake consensus algorithms: A systematic review. Journal of Environmental Management, 326, 116530.

[21] Alfriehat, N., Anbar, M., Aladaileh, M., Hasbullah, I., Shurbaji, T. A., Karuppayah, S., & Almomani, A. (2024). RPL-based attack detection approaches in IoT networks: review and taxonomy. Artificial Intelligence Review, 57(9), 248.

[22] Haseeb, S., Khalil Afzal, M., Tahir, M., Raza Jafri, M., & Raza, N. (2023). Energy-efficient selection of relay for UWSNs in the Internet of underwater things. *International Journal of Communication Systems*, *36*(18), e5619.

[23] Ran, X., Suyaroj, N., Tepsan, W., Ma, J., Zhou, X., & Deng, W. (2024). A hybrid genetic-fuzzy ant colony optimization algorithm for automatic K-means clustering in urban global positioning system. *Engineering Applications of Artificial Intelligence*, *137*, 109237.

[24] Saiyed, M. F., & Al-Anbagi, I. (2024). A Genetic Algorithm-and t-Test-Based System for DDoS Attack Detection in IoT Networks. *IEEE Access*, *12*, 25623-25641.

[25] Althiyabi, T., Ahmad, I., & Alassafi, M. O. (2024). Enhancing IoT Security: A Few-Shot Learning Approach for Intrusion Detection. *Mathematics*, *12*(7), 1055.

[26]“DDoS 2019 | Datasets | Research | Canadian Institute for Cybersecurity | UNB,” *www. unb. ca*. https: //www. unb. ca/cic/datasets/ddos-2019. html.

[27] Arefin, S., Chowdhury, M., Parvez, R., Ahmed, T., Abrar, A. S., & Sumaiya, F. (2024, May). Understanding APT detection using Machine learning algorithms: Is superior accuracy a thing?. In *2024 IEEE International Conference on Electro Information Technology (eIT)* (pp. 532-537). IEEE.

[28] Ali, J., Iwendi, C., Shan, G., Wu, H. C., Alenazi, M. J., Faheem, Z. B., & Biamba, C. N. (2024). Performance Optimization of Software-Defined Industrial Internet-of-Things (SD-IIoT). *IEEE Access*.

[29] Sinha, S. K. , Kumari, S. , Kataria, A. , Thangarasu, N. , & Sahoo, G. S. (2023). Blockchain empowerment: Investigating integration with software-defined networks and its impact on IoT privacy. *Multidisciplinary Reviews*, *6*.

[30] Kumaran, U. , Swetha, T. , Sharma, I. , Keerthana, R. , Tejashwini, V. , & Devarajan, G. G. (2024, May). Enhancing Cloud IoT Security With Blockchain and SDN. In *2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE)* (pp. 973-978). IEEE.

[31] Das, D. , Ghosh, U. , Evans, N. , & Shetty, S. (2024, June). Blockchain-Enabled Secure Device-to-Device Communication in Software-Defined Networking. In *2024 IEEE International Conference on Communications Workshops (ICC Workshops)* (pp. 1450-1455). IEEE.

پاورقی‌ها:

[1] Internet Of Thing(IOT)
[2] Sensor
[3] BlockChain
[4] Software Defined Network(SDN)
[5] Cloud Computing
[6] Whale Optimization Algorithm(WOA)
[7] Diffie–Hellman
[8] Proof of Stake (PoS)
[9] Proof of Work (PoW)
[10] Validators
[11] Ant Colony Optimization(ACO)
[12] Genetic Algorithm(GA)
[13] Routing Protocol for Low-Power and Lossy Networks
[14] Common Address Redundancy Protocol
[15] Network Bit Rate