

PERBANDINGAN UNJUK KERJA COAP DAN HTTP PADA TRANSAKSI DATA PERANGKAT IoT

Hendro F J Lami¹, Stephe I Pella²

^{1,2}) Program Studi Teknik Elektro Fakultas Sains dan Teknik Universitas Nusa Cendana

Email: h.lami@staf.undana.ac.id, s.i.pella@staf.undana.ac.id

Info Artikel

Histori Artikel:

Diterima Sep 13, 2022

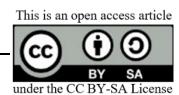
Direvisi Sep 28, 2022

Disetujui Okt 29, 2022

ABSTRACT

Internet of Things (IoT) is the interconnection of several devices that have specific tasks such as sensing and control. The services in the system can be in the form of one device - one service, many devices - one service, one device - many services, and many devices - many services. A communication protocol is agreed upon by the devices in the IoT system to ensure the success of data transactions. These protocols include MQTT, HTTP, AMQP, XMPP, Web Shocket, and COAP. This study compares the performance of the COAP and HTTP protocols in consideration of the number of transactions and energy consumption. The scenario used is to send RSSI data and battery level. The scenarios also use two modes of transmission power: 14dBm and 20dBm. The results show that the COAP protocol is 25 times more efficient than the HTTP protocol for a 20dBm transmit power scenario, while for a transmit power of 14dBm COAP is 61 times

Keywords: COAP, HTTP, IoT



ABSTRAK

Internet of Things (IoT) merupakan interkoneksi beberapa perangkat yang memiliki tugas tertentu misalnya penginderaan dan control. Layanan dalam sistem tersebut dapat berupa satu perangkat satu layanan, multi perangkat satu layanan, satu perangkat multi layanan, dan multi perangkat multi layanan. Sebuah protokol komunikasi disepakati oleh perangkat dalam sistem IoT untuk menjamin keberhasilan transaksi data. Protokol tersebut antara lain MQTT, HTTP, AMQP, XMPP, Web Shocket, dan COAP. Penelitian ini membandingkan unjuk kerja protokol COAP dan HTTP dalam hubungannya dengan jumlah transaksi dan konsumsi energi. Skenario yang digunakan adalah dengan melakukan pengiriman data rssi dan level daya baterai. Pengujian dilakukan dengan membedakan daya transmisi perangkat sebesar 14dBm dan 20dBm. Hasil pengujian, protokol COAP lebih hemat 25 kali dari protokol HTTP untuk scenario daya transmit sebesar 20dBm sedangkan untuk daya transmit sebesar 14dBm COAP lebih hemat 61 kali dibandingkan dengan protokol HTTP.

Kata Kunci: COAP, HTTP, IoT

Penulis Korespondensi:

Hendro F J Lami,

Program Studi Teknik Elektro Fakultas Sains dan Teknik,

Universitas Nusa Cendana,

Jalan Adisucipto – Penfui Kupang.

h.lami@staf.undana.ac.id



1. PENDAHULUAN

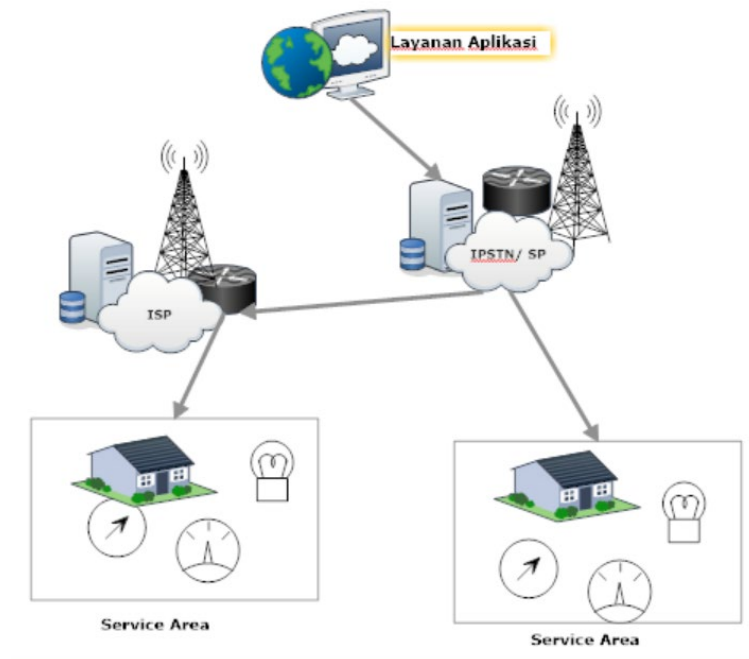
Internet of Things (IoT) memberikan kemampuan konektivitas antar perangkat untuk memberikan dukungan layanan tertentu. Layanan tersebut dapat berupa kemampuan pengindraan, monitoring dan pengontrolan[1]. Terlihat pada tabel 1, saat memberikan layanan, konektivitas perangkat IoT dapat terhubung menurut hubungan satu perangkat satu layanan, satu perangkat multi layanan, banyak perangkat satu layanan dan banyak perangkat multi layanan[2, 3].

Tabel 1 Layanan IoT

Konektivitas	Layanan	Perangkat IoT
Satu Perangkat-Satu layanan[4, 5]	EEG, EMG	Node MCU
Satu Perangkat-Multi Layanan[6]	Smart Band SoS system	Arduino+Esp8266 v01

Konektivitas	Layanan	Perangkat IoT
Multi Perangkat-Satu Layanan[7, 8]	Smart House, Surveillance System	Arduino+Esp8266 Esp8266-Raspberry Pi
Multi Perangkat-Multi Layanan[9]	Home Management and Controlling-Cloud IoT	Arduino Uno dan Esp8266

Berdasarkan jarak konektivitas suatu perangkat IoT yang memiliki layanan tertentu terbagi atas dua yaitu jarak dekat dan jarak jauh[10]. Komunikasi antara perangkat dapat menggunakan infrastruktur publik ataupun dapat mengembangkan interkoneksi non infrastruktur. Jaringan akses umum misalnya 5G, 4G, dan NB-IoT serta jaringan non infrastruktur misalnya Lorawan mampu mendukung konektivitas antar perangkat yang dapat membentuk sebuah jaringan distribusi data misalnya content distribution network[11-14].



Gambar 1 Content Distributed Network Model

Beberapa protokol komunikasi dapat diterapkan pada gambar 1 untuk mendukung transaksi data antar perangkat service area hingga dapat diakses oleh user. Protokol-protokol tersebut antara lain MQ Telemetry Transport (MQTT), Hypertext Transfer Protocol (HTTP), Advanced Message Queuing Protocol (AMQP), Extensible Messaging Presence Protocol (XMPP), Web Socket dan Constrained Application Protocol (COAP). Masing-masing protokol tersebut memiliki karakteristik dan keunggulannya masing-masing oleh karena itu pada

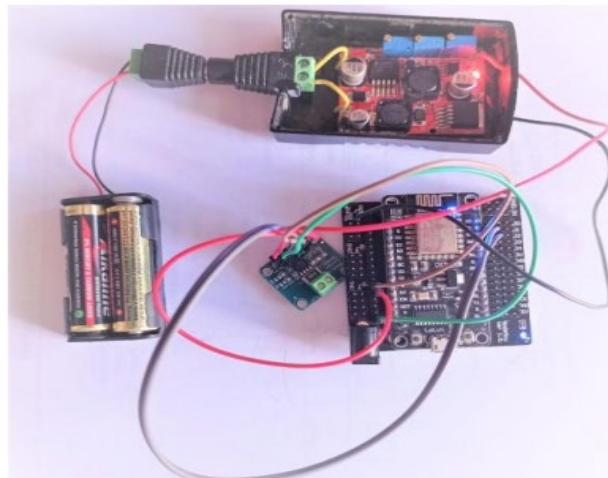
penelitian ini akan membandingkan unjuk kerja dari dua protokol terpilih yaitu COAP dan HTTP dengan menggunakan perangkat esp8266 dan sensor.

2. METODE PENELITIAN

Penelitian ini menggunakan dua buah perangkat esp8266 sebagai sensor node yang terkoneksi dengan sebuah sensor arus INA219. Dua buah baterai 1.5V menjadi catu daya bagi perangkat sensor node. Selain itu perangkat sensor node tersebut juga terkoneksi dengan sebuah converter

DC-to-DC XL6009 sehingga memungkinkan sensor node menjadi perangkat portable namun tetap menjamin tegangan input pada esp8266 berada pada level 5V. Skenario pada penelitian ini adalah salah satu sensor node akan mengirimkan informasi

mengenai status level tegangan catu serta informasi mengenai status receives signal strength indicating terhadap sensor node pemancar. Gambar 2 memberikan gambaran konfigurasi perangkat pada sisi sensor node.



Gambar 2 Konfigurasi Perangkat Sensor Node Esp8266

Perbandingan kehandalan antara protokol CoAP dan HTTP dilakukan menggunakan sensor node yang sama dengan data transmisi berupa RSSI dan level daya baterai. Analisa dilakukan dengan membandingkan Round trip time dari data yang diterima menggunakan protokol CoAP terhadap paket Round trip time dari data yang diterima menggunakan protokol HTTP. Round trip time merupakan waktu yang diperlukan dari saat client mengirim request sampai client menerima respons dari server. Pengukuran dilakukan dengan menggunakan aplikasi Wireshark yang berjalan pada jaringan yang sama dengan client dan server untuk mengukur Round trip time saat transaksi data berjalan. Berikut ini adalah potongan kode program untuk mendapatkan level daya catu dan RSSI

```
long rssi = WiFi.RSSI();
busvoltage = ina219.getBusVoltage_V();
shuntvoltage = ina219.getShuntVoltage_mV();
loadvoltage = busvoltage + (shuntvoltage / 1000);
level = ((loadvoltage-Vmin) / (Vmax-Vmin)) * 100
```

Untuk COAP server potongan kodenya sebagai berikut:

```
server.CreateResource("RSSI_&_Battery",
Thing::CoAP::ContentFormat::TextPlain, true)
.OnGet([](Thing::CoAP::Request & request) { result = ""
return Thing::CoAP::Status::Content(result); })
```

Untuk HTTP server potongan kode programnya sebagai berikut:

```
WiFiClient client = server.available(); if (client)
{ Serial.println("New Client."); String
currentLine = ""; currentTime = millis();
previousTime = currentTime; while
(client.connected()) && currentTime - previous
```

Round trip time (*RTT*) data transmisi mempengaruhi seberapa lama sensor node portable mampu bertahan dalam suatu jaringan IoT. Pengujian tersebut menggunakan dua mode daya transmisi yaitu 14dBm dan 20 dBm dimana pemilihan daya tersebut berdasarkan pertimbangan daya transmit esp8266 [15]. Melalui persamaan (1) perbandingan konsumsi energi baterai terkecil antara COAP dan HTTP dapat diketahui.

$$W = P \times t \tag{1}$$

Keterangan:

W = energi (Joules)

P = Daya (Watt = Joules/ second)

t = waktu (second)

3. HASIL DAN PEMBAHASAN

Melalui pengujian skala lab, berikut hasil implementasi dan pengujian dari kedua protokol tersebut. Untuk pengujian yang pertama adalah untuk mendapatkan round trip time transaksi data melalui *Wireshark* seperti terlihat pada gambar sedangkan hasilnya terlihat pada tabel

Tabel 1 RTT dan Konsumsi Energi Protokol COAP

Paket	Daya Transmisi +14dBm		Daya Transmisi +20 dBm		
	RTT (detik)	Energi (WH)	Paket	RTT (detik)	Energi (WH)
1	0.007816	5.4536 × 10 ⁻⁸	1	0.106212	2.9503 × 10 ⁻⁶
2	0.042024	2.9322 × 10 ⁻⁷	2	0.121837	3.3844 × 10 ⁻⁶
3	0.007474	5.215 × 10 ⁻⁸	3	0.063843	1.7734 × 10 ⁻⁶

Daya Transmisi +14dBm			Daya Transmisi +20 dBm		
4	0.047467	3.312×10^{-7}	4	0.086787	2.4108×10^{-6}
5	0.070099	4.891×10^{-7}	5	0.076181	2.1161×10^{-6}
Rerata	0.034976	2.4404×10^{-7}	Rerata	.090972	2.527×10^{-6}

RTT protokol CoAP dalam satu kali transaksi data menggunakan standar 802.11n pada daya transmit +14dBm adalah 0.034976 second dengan konsumsi energi $2.4404 \times 10^{-7}Wh$ dan pada daya transmit +20dBm mendapat RTT sebesar 0.090972 second dan konsumsi energi $2.527 \times 10^{-6}Wh$.

Dengan menggunakan metode yang sama untuk menghitung konsumsi daya pada protokol CoAP, dari hasil capture Wireshark, RTT pada protokol HTTP dihitung dari pesan SYN pada saat pembentukan koneksi TCP dikirim *client* sampai pesan ACK untuk memutuskan hubungan TCP diterima. Hasil perhitungan konsumsi energi pada protokol HTTP berdasarkan RTT pada setiap paket dan pada rata-rata RTT yang diperoleh pada daya transmisi +14 dBm dan daya transmisi +20 dBm dapat dilihat seperti pada tabel 2.

Tabel 2 RTT dan Konsumsi Energi Protokol HTTP

Daya Transmisi +14dBm			Daya Transmisi +20 dBm		
Paket	RTT (detik)	Energi (WH)	Paket	RTT (detik)	Energi (WH)
1	2.163715	1.5097×10^{-5}	1	2.220679	6.1686×10^{-5}
2	2.147898	1.4987×10^{-5}	2	2.186707	6.0742×10^{-5}
3	2.233671	1.4987×10^{-5}	3	2.386452	6.629×10^{-5}

Daya Transmisi +14dBm			Daya Transmisi +20 dBm		
4	2.159739	1.5069×10^{-5}	4	2.386452	6.533×10^{-5}
5	2.173076	1.5163×10^{-5}	5	2.166680	6.018×10^{-5}
Rerata	2.17562	1.518×10^{-5}	Rerata	2.262483	6.2847×10^{-5}

RTT protokol HTTP dalam satu kali transaksi data menggunakan standar 802.11n pada daya transmit +14dBm adalah 2.17562 second dengan konsumsi energi $1.518 \times 10^{-5}Wh$ dan pada daya transmit +20dBm mendapat RTT sebesar 2.262483 second dan konsumsi energi $6.2847 \times 10^{-5}Wh$.

Baterai alkaline AA memiliki kapasitas sebesar 3000mAh atau sama dengan 3Ah dan tegangan 1.5v[16] sehingga kapasitas energi sama dengan 4.5Wh untuk satu baterai. Jika, request dilakukan setiap 30 menit di dapat 48 transaksi /hari. Berdasarkan perhitungan konsumsi energi dalam satu kali transaksi data maka dengan mengabaikan konsumsi energi lain pada sensor node untuk menghidupkan Node Mcu Esp 8266, keadaan standby, sensing dan prosesi, maka lama penggunaan baterai dapat di tentukan sebagai berikut:

CoAP +14 dBm:

Penggunaan energi 1 kali transaksi = $2.4404 \times 10^{-7}Wh$ Maka, jumlah transaksi n yang mungkin:

$$n = \frac{4.5WH}{2.4404 \times 10^{-7}Wh} = 18439600 \text{ transaksi}$$

CoAP +20 dBm:

Penggunaan energi 1 kali transaksi = $2.527 \times 10^{-6}Wh$ Maka, jumlah transaksi n yang mungkin:

$$n = \frac{4.5WH}{2.527 \times 10^{-6}Wh} = 1780767 \text{ transaksi}$$

RTT					
43	5.056409	192.168.43.121	192.168.43.224	CoAP	64 CON, MID:63994, GET, Block #0, /RSSI & Battery
44	5.102800	192.168.43.224	192.168.43.121	CoAP	82 ACK, MID:63994, 2.05 Content (text/plain)
45	6.384921	192.168.43.121	192.168.43.224	CoAP	64 CON, MID:63995, GET, Block #0, /RSSI & Battery
46	6.435344	192.168.43.224	192.168.43.121	CoAP	82 ACK, MID:63995, 2.05 Content (text/plain)

RTT					
43	*REF*	192.168.43.121	192.168.43.224	CoAP	64 CON, MID:63994, GET, Block #0, /RSSI & Battery
44	0.046391	192.168.43.224	192.168.43.121	CoAP	82 ACK, MID:63994, 2.05 Content (text/plain)
45	*REF*	192.168.43.121	192.168.43.224	CoAP	64 CON, MID:63995, GET, Block #0, /RSSI & Battery
46	0.050423	192.168.43.224	192.168.43.121	CoAP	82 ACK, MID:63995, 2.05 Content (text/plain)

Gambar 3 Hasil Sniffing Protocol COAP

RTT					
833	*REF*	192.168.43.121	192.168.43.68	TCP	66 52704 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
834	0.099950	192.168.43.68	192.168.43.121	TCP	54 80 → 52701 [ACK] Seq=2 Ack=2 Win=2143 Len=0
835	0.100726	192.168.43.68	192.168.43.121	TCP	62 80 → 52703 [SYN, ACK] Seq=0 Ack=1 Win=2144 Len=0 MSS=536 SACK_PERM=1
836	0.100726	192.168.43.68	192.168.43.121	TCP	62 80 → 52704 [SYN, ACK] Seq=0 Ack=1 Win=2144 Len=0 MSS=536 SACK_PERM=1
837	0.100923	192.168.43.121	192.168.43.68	TCP	54 52703 → 80 [ACK] Seq=1 Ack=1 Win=65392 Len=0
838	0.101001	192.168.43.121	192.168.43.68	TCP	54 52704 → 80 [ACK] Seq=1 Ack=1 Win=65392 Len=0
839	0.101903	192.168.43.121	192.168.43.68	HTTP	534 GET /RSSI&BAT HTTP/1.1
840	0.221212	192.168.43.68	192.168.43.121	TCP	54 80 → 52704 [ACK] Seq=1 Ack=481 Win=1664 Len=0
843	2.143315	192.168.43.68	192.168.43.121	TCP	54 80 → 52703 [FIN, ACK] Seq=1 Ack=1 Win=2144 Len=0
844	2.143432	192.168.43.121	192.168.43.68	TCP	54 52703 → 80 [ACK] Seq=1 Ack=2 Win=65392 Len=0
845	2.146100	192.168.43.68	192.168.43.121	TCP	69 80 → 52704 [PSH, ACK] Seq=1 Ack=481 Win=1664 Len=15 [TCP segment of a reassembled PDU]
846	2.150944	192.168.43.68	192.168.43.121	TCP	590 80 → 52704 [PSH, ACK] Seq=16 Ack=481 Win=1664 Len=536 [TCP segment of a reassembled PDU]
847	2.151158	192.168.43.121	192.168.43.68	TCP	54 52704 → 80 [ACK] Seq=481 Ack=552 Win=65392 Len=0
848	2.155307	192.168.43.68	192.168.43.121	TCP	276 80 → 52704 [PSH, ACK] Seq=552 Ack=481 Win=1664 Len=222 [TCP segment of a reassembled PDU]
849	2.202736	192.168.43.121	192.168.43.68	TCP	54 52704 → 80 [ACK] Seq=481 Ack=774 Win=65170 Len=0
850	2.207888	192.168.43.68	192.168.43.121	HTTP	54 HTTP/1.1 200 OK (text/html)
851	2.207940	192.168.43.121	192.168.43.68	TCP	54 52704 → 80 [ACK] Seq=481 Ack=775 Win=65170 Len=0
852	2.208115	192.168.43.121	192.168.43.68	TCP	54 52704 → 80 [FIN, ACK] Seq=481 Ack=775 Win=65170 Len=0
853	2.212336	192.168.43.68	192.168.43.121	TCP	54 80 → 52704 [ACK] Seq=775 Ack=482 Win=1663 Len=0

Gambar 4 Hasil Sniffing Protocol HTTP

HTTP +14 dBm:

Penggunaan energi 1 kali transaksi = 1.518×10^{-5} Wh
 Maka, jumlah transaksi n yang mungkin:

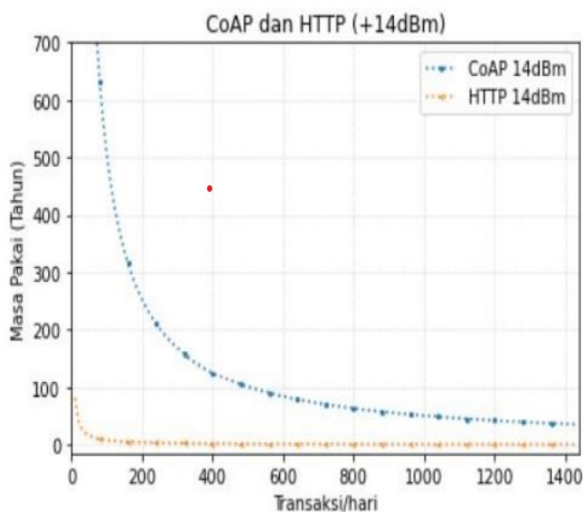
$$n = \frac{4.5WH}{1.518 \times 10^{-5}Wh} = 1780767 \text{ transaksi}$$

HTTP +20 dBm:

Penggunaan energi 1 kali transaksi = 6.2847×10^{-5} Wh
 Maka, jumlah transaksi n yang mungkin:

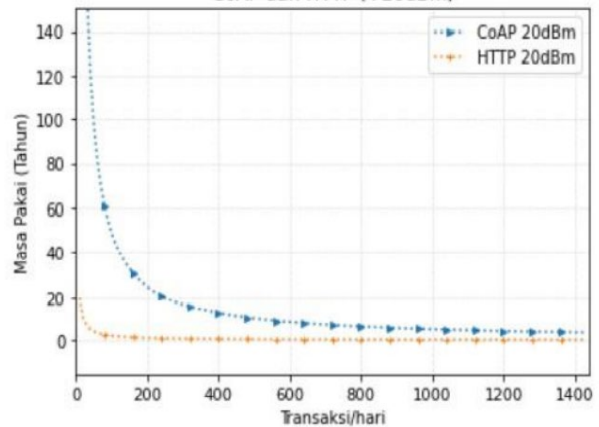
$$n = \frac{4.5WH}{6.2847 \times 10^{-5}Wh} = 71602 \text{ transaksi}$$

Berdasarkan hasil perhitungan masa pakai pada daya +14 dBm protokol CoAP 61 kali lebih hemat energi dibandingkan protokol HTTP dan pada daya +20 dBm protokol CoAP 25 kali lebih hemat energi dibandingkan protokol HTTP. Hubungan antara daya transmisi dan konsumsi energi terhadap total transaksi terlihat pada gambar 4 dan gambar 5.



Gambar 5 Masa Pakai Baterai Terhadap Perangkat Esp8266 Daya 14dbm

CoAP dan HTTP (+20dBm)



Gambar 6 Masa Pakai Baterai Terhadap Perangkat Esp8266 Daya 20dbm

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan dua protokol IoT yaitu COAP dan HTTP untuk melihat unjuk kerja kedua protokol tersebut. Berdasarkan hasil pengujian, protokol COAP lebih hemat 25 kali dari protokol HTTP untuk scenario daya transmit sebesar 20dBm sedangkan untuk daya transmit sebesar 14dBm COAP lebih hemat 61 kali dibandingkan dengan protokol HTTP.

Penerapan daya transmit maksimum esp8266 sebesar 20dBm memberikan hasil penggunaan energi untuk satu kali transaksi pada protokol COAP adalah sebesar 2.4404×10^{-7} Wh. Sementara itu konsumsi energi untuk satu kali transaksi protokol HTTP adalah sebesar 1.518×10^{-5} Wh.

Penerapan daya transmit 14dBm memberikan hasil penggunaan energi untuk satu kali transaksi pada protokol COAP adalah sebesar 2.4404×10^{-7} Wh. Sementara itu konsumsi energi untuk satu kali transaksi protokol HTTP adalah sebesar 1.518×10^{-5} Wh.

Penerapan daya transmit 20dBm memberikan hasil penggunaan energi untuk satu kali transaksi pada protokol COAP adalah sebesar 2.527×10^{-7} Wh.

Sementara itu konsumsi energi untuk satu kali transaksi protokol HTTP adalah sebesar 6.2847×10^{-5} Wh.

DAFTAR PUSTAKA

- [1] A. Verma, S. Prakash, V. Srivastava, A. Kumar, and S. C. Mukhopadhyay, "Sensing, controlling, and IoT infrastructure in smart building: A review," *IEEE Sensors Journal*, vol. 19, no. 20, pp. 9036-9046, 2019.
- [2] S. K. Datta and C. Bonnet, "Connect and control things: integrating lightweight IoT framework into a mobile application," in *2015 9th international conference on next generation mobile applications, services and technologies*, 2015, pp. 66-71: IEEE.
- [3] F. Samie, L. Bauer, and J. Henkel, "IoT technologies for embedded computing: A survey," in *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ ISSS)*, 2016, pp. 1-10: IEEE.
- [4] A. Kelati, I. B. Dhaou, and H. Tenhunen, "Biosignal monitoring platform using Wearable IoT," in *Proceedings of the 22st Conference of Open Innovations Association FRUCT, Petrozavodsk, Russia*, 2018, pp. 9-13.
- [5] A. A. Abdellatif, M. G. Khafagy, A. Mohamed, and C.-F. J. I. I. o. T. J. Chiasserini, "EEG-based transceiver design with data decomposition for healthcare IoT applications," vol. 5, no. 5, pp. 3569-3579, 2018.
- [6] A. Karmakar, K. Ganguly, and P. S. Banerjee, "SafeBand: IoT-Based Smart Security Band with Instant SOS Messaging," in *Proceedings of International Conference on Advanced Computing Applications*, 2022, pp. 127-140: Springer.
- [7] M. S. Mahamud, M. S. R. Zishan, S. I. Ahmad, A. R. Rahman, M. Hasan, and M. L. Rahman, "Domicile-an IoT based smart home automation system," in *2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, 2019, pp. 493-497: IEEE.
- [8] A. K. Gupta and R. Johari, "IOT based electrical device surveillance and control system," in *2019 4th international conference on internet of things: Smart innovation and usages (IoT-SIU)*, 2019, pp. 1-5: IEEE.
- [9] M. M. A. Zahra, M. J. Mohsin, L. A. J. P. o. E. Abdul-Rahaim, and N. Sciences, "Artificial intelligent smart home automation with secured camera management-based GSM, cloud computing, and arduino," vol. 8, no. 4, pp. 2160-2168, 2020.
- [10] J. Ding, M. Nemati, C. Ranaweera, and J. J. a. p. a. Choi, "IoT connectivity technologies and applications: A survey," 2020.
- [11] S. Li, L. Da Xu, and S. J. J. o. I. I. I. Zhao, "5G Internet of Things: A survey," vol. 10, pp. 1-9, 2018.
- [12] S. Popli, R. K. Jha, and S. J. I. A. Jain, "A survey on energy efficient narrowband internet of things (NB-IoT): architecture, application and challenges," vol. 7, pp. 16739-16776, 2018.
- [13] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. J. S. Hoebeke, "A survey of LoRaWAN for IoT: From technology to application," vol. 18, no. 11, p. 3995, 2018.
- [14] S. K. Lee, M. Bae, and H. Kim, "Future of IoT Networks: A Survey," vol. 7, no. 10, p. 1072, 2017.
- [15] B. D. Costa and G. Shet, "Low Power IoT System Node Design," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2019, pp. 1-8: IEEE.
- [16] M. F. Almeida, S. M. Xará, J. Delgado, and C. A. J. W. M. Costa, "Characterization of spent AA household alkaline batteries," vol. 26, no. 5, pp. 466-476, 2006.