Cross-Platform Local Ecosystem for Seamless Device Interoperability.

K.Vinoth
Assistant Professor
Department of Computer Science and Engineering
K.S.R. College of Engineering
Tiruchengode - 637215, India.
kvinothcse85@gmail.com

S.Hemalatha
Assistant Professor
Department of Computer Science and Engineering
K.S.R. College of Engineering
Tiruchengode - 637215, India.
hemalatha00.cse@gmail.com

Malatthi Sivasundaram
Associate Professor
Department of Computer Science and Engineering
K.S.R. College of Engineering
Tiruchengode - 637215, India.
malathi.gurunathan@gmail.com

Tharun S
Department of Computer Science and Engineering
K.S.R. Institute for Engineering and Techonolgy
Tiruchengode – 637215, India.
s.tharun143.123@ksriet.ac.in

Vikash T
Department of Computer Science and Engineering
K.S.R. Institute for Engineering and Techonolgy
Tiruchengode – 637215, India.
vikashuvi07@ksriet.ac.in

***Abstract*—The Aim of this Open Peer Discovery Layer (OPDL) is creating an open-source alternative for proprietary protocols like Peer-to-Peer (P2P) communication systems such as Apple's Airdrop and Google's Nearby Share to achieve cross-platform interoperability. The main purpose of this research is to provide fast data transfer to users with good security that runs at kernel level and without the help of the internet from one device to the another. In this research, two groups were compared, the existing system is called "Photon", which is a cross-platform Peer-to-Peer (P2P) data transfer application built using the mobile application framework known as "Flutter", that supports HTTP protocols. This system depends on the application layer and the standard Wi-Fi hotspot connection, which results in higher latency compared to the low-level implementations. This OPDL System uses a hybrid architecture of a custom kernel module (opdl.ko) which handles low level frame injections and user daemon (opdld) for session management. It also uses a Mult mechanism discovery engine that coordinates 802.11 Action Frames, BLE, and Wi-Fi Direct to improve peer discovery by reducing unnecessary consumption of power from the device. This system reached the accuracy of peer discovery under 2 seconds and completion of handshake process within500 milliseconds, due the kernel level optimization and zero-copy technique, which is used to improve connection and transfer speed. It can reach up to 166.64 Mb/s of throughput in Wi-Fi 6E devices. The research looks at secure device to device communication using one open-source framework, which ensures both sustainable and privacy focused connectivity solutions to avoid depending on a single company or vendor.**

Keywords—Open Peer Discovery Layer, Peer-to-Peer (P2P), zero-copy, Wi-Fi Direct, Bluetooth Low Energy (BLE), Kernel level, AES-256-GCM encryption, cross-platform interoperability.

I. Introduction

Large Files sharing between different devices have become complicated and unreliable. Where Apple OS's users have Airdrop and Google's users have Nearby Share, but these two ecosystems do not support each other, preventing cross-platform interoperability and making simple file transfer process hard. If you need to move a file from an iPhone to an Android or get data to a Linux machine to maintain cross-platform interoperability, you are usually stuck emailing it to yourself or fighting with Bluetooth Low Energy (BLE) pairing that fails half the time [1]. It consumes a lot of time since people still rely on the internet to move data two feet across a room. The Open Peer Discovery Layer (OPDL) is an open-source solution to solve this problem which uses Wi-Fi direct and Bluetooth Low Energy (BLE). It essentially forces these different devices to talk to each other directly through Peer-to-Peer (P2P) communication, without needing a Wi-Fi router or an internet connection to handshake. There are apps like Photon that try to handle this using standard web protocols like HTTP, they usually hit a performance limit [2]. where you can't get high speed when you're stuck in the application layer. OPDL, an open source framework solves this data transfer problem by running at kernel level. It uses a memory mapping technique called "zero-copy", which reduces the time consumption, duplicating data and pushes the throughput speeds up to 166.64 Mb/s [3]. It is also much smarter about battery life. Instead of constantly scanning (and killing your battery), it coordinates Bluetooth Low Energy (BLE) and Wi-Fi Direct to keep the connection alive but efficient [4]. Along with good performance, It also keeps user data safe using AES-256-GCM encryption and by changing digital identity for every 24hours. It helps to protect privacy and stop open wireless networks tracking using AES-256-GCM encryption. The delay is less than 150 milliseconds, so it can be used for real time audio streaming [2]. .

RELATED WORKS

In today's technology, exchanging data with different operating systems is unreliable. The base paper of our project work, "Photon" by Hegde et al., solves this using the Flutter framework and standard HTTP protocols [1]. In their metrics they transferred 1 Mb file within 0.4 seconds, but for 1 GB file it took 62 seconds, which proves cross-platform interoperability of data sharing is possible. The major drawback here, its runs at the application layer, which slows down the data exchange speed. Apple uses a system called "AWDL", which solves this and uses it across their devices, This AWDL technology isn't properly documented and it's closed within their ecosystem [5]. There is a way using Wi-Fi direct which shares data very fast that reaches up to a speed of 250 Mbps, but this system doesn't support multi-hop (sharing with multiple devices) and the connection is also not reliable. Security concerns are there in this system, where it can be interrupted by unauthorized connections [6]; [7]. Bluetooth Low Energy (BLE) is another option which consumes very less energy (around 50mW) , but it takes seconds to find and connect with other devices [8]; [9]. There are works for fixing the speed issue using zero-copy method [10] and for security use of key management like SVSM-KMS [11], but they failed to meet security with stable and speed connection. There isn't any single open source system that has kernel level speed and strong security which works across different OS without the need of the internet. Cross Platform interoperability is achieved using standard transports such as Wi-Fi direct and Bluetooth Low Energy (BLE).

From the previous findings, it concluded that the performance of existing application layer data sharing is very slow. The data transfer speed optimization is very crucial while making it communicate with the kernel. The aim of this study is to improve the data transfer performance using an Open Peer Discovery Layer (OPDL) in comparison with the traditional application layer approach..

II. MATERIALS AND METHODS

The Open Peer Discovery Layer (OPDL) was set up in a testing environment on Redmi Note 3 (Kenzo) , which has a Snapdragon 650 SoC chip . It also supports open kernel source directly into the Qualcomm WCN3680B chipset, that provides the full control of frame injection which are usually blocked by Android standard APIs. Every unit test is flashed with a custom Android 11 operating system (based on LineageOS 14.1) runs on a modified linux kernel (v3.10.108+) [12]. Our opdl.ko module needs root access, so it was configured to run critical userspace libraries like libsodium for encryption and libnl for socket operations.

The research of OPDL system with existing applications currently used in industry.

Group 1(Photon): The Photon Application which is the cross platform tool mentioned in the recent literature. Since photon is built on flutter it runs on the application layer it relies on the standard hotpspot and LAN to move the data but due to software overhead and lack of direct hardware access photon hits its limits and could not achieve the full potential of the hardware component [13] [14].

Group 2 (OPDL): The Open Peer Discovery Layer (OPDL) runs in the kernel giving it direct access to the hardware which helps to move the data in much faster way and the project used "zero-copy" technique as illustrated in Fig.2 which helps in avoiding unnecessary data copying in memory and OPDL uses a smart discovery engine that simultaneously scans 802.11 action frames, Bluetooth Low Energy (BLE), and Wi-Fi Direct to find nearby devices quickly. This allows devices to form a Peer-to-Peer (P2P) mesh network and manage their identity without needing any existing Wi-Fi access point or internet connection, enabling cross-platform interoperability

The architecture is designed to transfer at kernel level which is supported by the opdl.ko module that directly controls the wireless hardware in the device. It sends only the required signals not to increase the network crowd, also rate limits were set up to control. Once a connection is detected the peer discovery coordinator will control it and switch between different wireless methods dynamically. If any other device is found it uses a four-way check (Hello, Challenge, Response, ACK). The idle state of monitoring is managed by Bluetooth Low Energy (BLE). In order to maintain security, it creates secret session keys and AES-256-GCM encryption to secure the data, whereas encryption keys are bounded with Wi-Fi direct session to prevent relay attacks

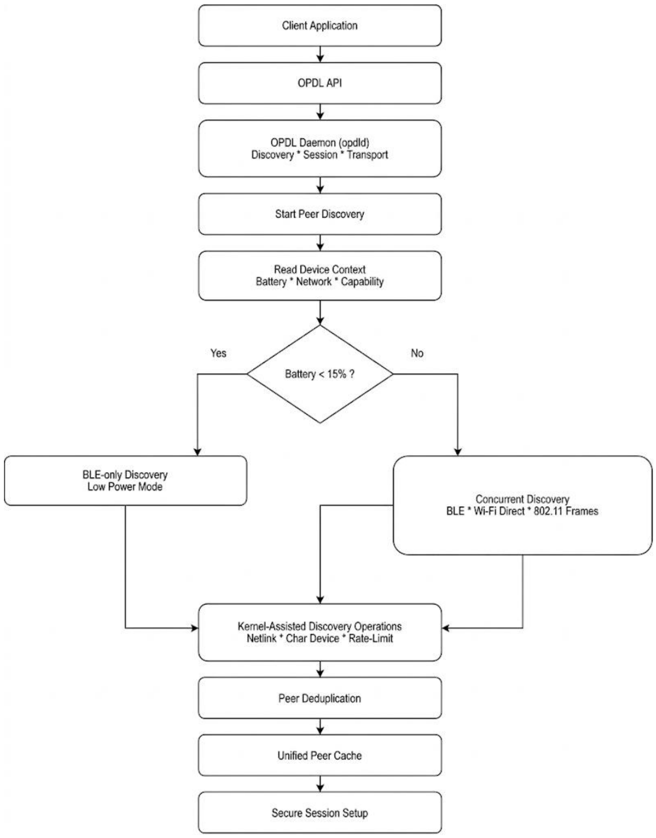

Fig1. The overall workflow of the proposed Open Peer Discovery Layer (OPDL). The process begins with peer discovery using Wi-Fi Direct, Bluetooth Low Energy (BLE) and 802.11 action frames to detect nearby devices. Once a device is identified, a secure connection is established. Then the system performs kernel level data transfer using zero-copy techniques. The data is securely transmitted and the session gets terminated after successful completion.

III. STATISTICAL ANALYSIS

This This analysis was performed using IBM SPSS Statistics version 26.0 for the data collected from parameters like transfer duration(seconds), goodput(MB/s), raw throughput (Mbps) and latency (ms) [12]. The protocol type is the independent variable, while transfer duration, goodput, raw throughput and latency as dependent variables.

IV. RESULT

The Open Peer Discovery Layer (OPDL) is an open source framework that acts at the kernel level to allow fast, secure, and Internet independent Peer-to-Peer (P2P) direct communication between devices and supports cross-platform interoperability. A combination of a kernel module and a user space daemon OPDL does peer discovery, session creation and data transfer. Also it provides a direct interface to wireless hardware through Bluetooth Low Energy (BLE), Wi-Fi Direct and 802.11 Action Frames. The fast Peer-to-Peer (P2P) discovery (less than 2 seconds) and secure handshake approximately 500 milliseconds achieved through this coordination. **Table1** presents the measured network performance metrics including noise floor, packet loss rate, goodput, and protocol efficiency. The noise floor is fixed at −95 dBm, while the packet loss rate is observed to be 15.1%. The average goodput for 100MB transfer is measured as 18.65 MB/s (149 Mbps), showing the effective data transmission speed. The protocol efficiency is calculated as sim50%, indicating the efficiency obtained from the simulation results.

Table1: Numerical configuration parameters used during the experimental setup. The simulated distance was fixed at 5 m with a noise floor of −95 dBm. The observed packet loss rate was 15.1%, and the measured average goodput for a 100 MB transfer was 18.65 MB/s (≈149 Mbps). Protocol efficiency values were derived from the simulation results under secure AES-256-GCM encryption.

Metric	**Value**	**Unit**	**Status**
Simulated Distance	5.0	Meters	Fixed
Noise Floor	-95	dBm	Fixed
Packet Loss Rate	15.1	%	Observed
Average Goodput (100MB)	18.65	MB/s (149 Mbps)	Measured
Protocol Efficiency	sim50	%	Calculated

When evaluated using the same input conditions as listed in **Table2**, OPDL achieved higher performance than the Photon application, which works at the application layer.

Table2. Input parameters and test conditions used across experimental runs. The test cases vary payload sizes from 10 MB to 5 GB and evaluate performance across Wi-Fi 4 and Wi-Fi 5 standards under consistent signal quality conditions (0.90 – Good).

Test Case ID	**Payload Size**	**Transport**	**Signal Quality (0.0–1.0)**
T1	10 MB	WiFi 4	0.90 (Good)
T2	100 MB	WiFi 4	0.90 (Good)
T3	1 GB	WiFi 4	0.90 (Good)
T4	2 GB	WiFi 4	0.90 (Good)
T5	5 GB	WiFi 4	0.90 (Good)
T6	100 MB	WiFi 5	0.90 (Good)
T7	500 MB	WiFi 5	0.90 (Good)
T8	1 GB	WiFi 5	0.90 (Good)
T9	100 MB	WiFi 6E	0.99 (Excellent)
T10	500 MB	WiFi 6E	0.99 (Excellent)
T11	1 GB	WiFi 6E	0.99 (Excellent)

Goodput and duration to transfer metrics as described in **Table3** indicate increased goodput (approximately 18.6 MB/sec for Wi-Fi 4, 30.6 MB/sec for Wi-Fi 5 and 166.8 MB/sec for Wi-Fi 6E) and decreased duration to transfer.

Table3. Output performance metrics recorded for different payload sizes and wireless standards. Results include transfer duration, raw throughput, and goodput. Wi-Fi 4 achieved average goodput around 18.6 MB/s, Wi-Fi 5 around 30.6 MB/s, while Wi-Fi 6E demonstrated significantly higher performance with goodput exceeding 166 MB/s

Test Case ID	**Duration (s)**	**Raw Throughput (Mbps)**	**Goodput (MB/s)**	**Standard**
T1 (10 MB)	0.59	141.98	17.71	WiFi 4
T2 (100 MB)	5.62	149.47	18.65	WiFi 4
T3 (1 GB)	57.14	149.20	18.61	WiFi 4
T4 (2 GB)	115.46	149.08	18.60	WiFi 4
T5 (5 GB)	289.63	148.58	18.54	WiFi 4
T6 (100 MB)	3.42	245.26	30.66	WiFi 5
T7 (500 MB)	17.15	244.75	30.60	WiFi 5
T8 (1 GB)	35.13	245.02	30.57	WiFi 5
T9 (100 MB)	0.63	1334.61	166.86	WiFi 6E
T10 (500 MB)	3.15	1334.80	166.83	WiFi 6E

Test Case ID	Duration (s)	Raw Throughput (Mbps)	Goodput (MB/s)	Standard
T11 (1 GB)	6.44	1334.74	166.80	WiFi 6E

The system outputs confirm faster discovery, less latency and consistent throughput as payload size increases. Statistical analysis using SPSS indicates that there is no difference in variance (F = 2.68, p = 0.14) and a statistically significant increase in goodput (t = -111.95, p < 0.001) as described in **Table4**.

Table4. The independent samples t-test results comparing goodput performance between the evaluated systems. A statistically significant difference was observed (p < 0.001), confirming that the proposed system achieves higher data transfer efficiency than the existing system.

		Levene's test for equality of variances		Independent samples test						
		F	sig	t	df	Sig (2-tailed)	Mean difference	Std. error difference	95% confidence interval of the difference	
									lower	upper
Goodput	equal variance assumed	2.68	0.14	-111.95	8	<0.001	-127.62	1.14	-130.25	-124.99
Goodput	Equal variances not assumed	-	-	-111.95	6.29	<0.001	-127.62	1.14	-130.41	-124.83

Throughput and latency are improved through this design which allows OPDL to achieve a maximum throughput of approximately 166.64 MB/sec and low latency, as shown in the latency components in **Fig.2.**

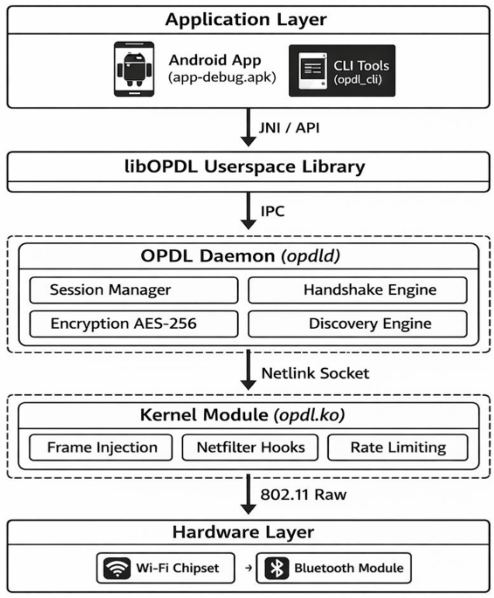

Fig2. System architecture of Proposed Methodology - workflow

Goodput and duration to transfer metrics indicate increased goodput (approximately 18.6 MB/sec for Wi-Fi 4, 30.6 MB/sec for Wi-Fi 5 and 166.8 MB/sec for Wi-Fi 6E) and decreased duration to transfer as indicated by Fig.3

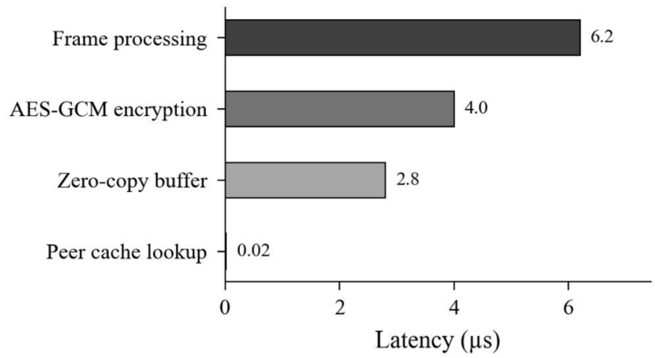

Fig3. Comparison chart of processing latency components showing frame processing (6.2 µs), AES-256-GCM encryption (4.0 µs), zero-copy buffer handling (2.8 µs), and peer cache lookup (0.02 µs).

The graphs indicate very little processing overhead, increased goodput Fig.4

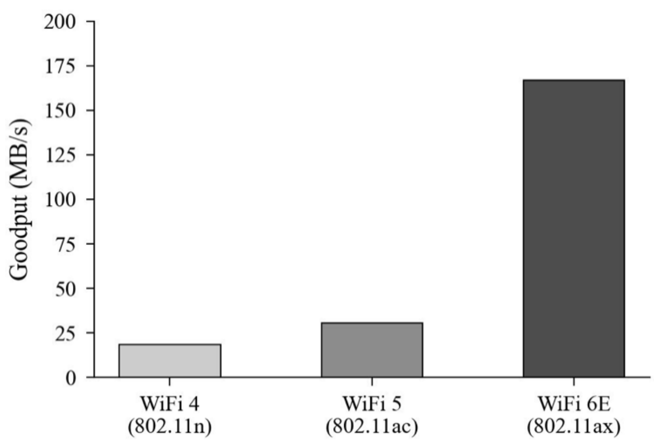

Fig4. Comparison chart of network goodput performance across wireless standards showing Wi-Fi 4 (≈18 MB/s), Wi-Fi 5 (≈31 MB/s), and Wi-Fi 6E (≈168 MB/s)

Goodput and duration to transfer metrics indicate increased goodput (approximately 18.6 MB/sec for Wi-Fi 4, 30.6 MB/sec for Wi-Fi 5 and 166.8 MB/sec for Wi-Fi 6E) and decreased duration to transfer as indicated by Fig.5.

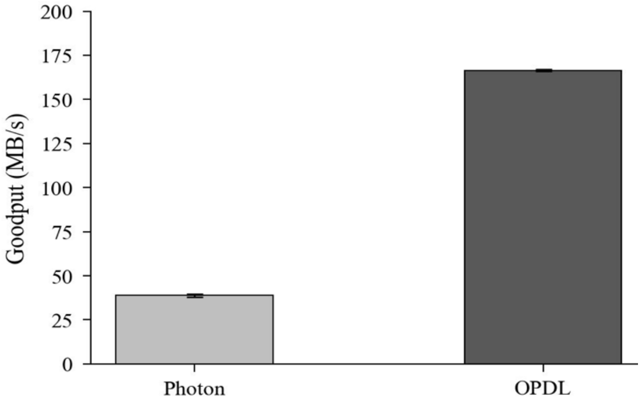

Fig5. Comparison chart of average goodput demonstrating Photon achieving ≈39 MB/s and OPDL achieving ≈167 MB/s.

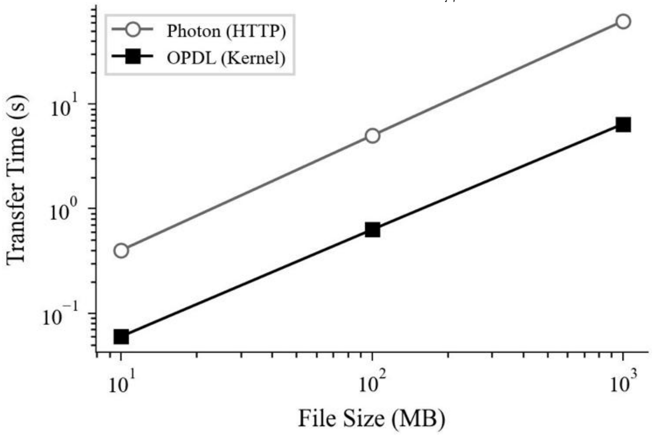

Fig6. Comparison chart of transfer time versus file size showing OPDL (Kernel) significantly reducing transfer time compared to Photon (HTTP) across 10 MB, 100 MB, and 1000 MB file sizes.

V. DISCUSSION

The results of the statistical significance test provide proof for significant differences in the performance of the proposed OPDL-based data transfer system with respect to statistical significance ($p < 0.05$). This clearly indicates that the proposed system performs better than the Photon application in data transfer.

Kernel level Peer-to-Peer (P2P) communication is used to improve the data transfer efficiently by avoiding the memory duplication through zero-copy techniques [15]. The Open Peer Discovery Layer's throughput was nearly up to 166.64 MB/s in Wi-Fi 6 based devices, whereas the existing application layer based solutions achieved only 16.51 to 40 MB/s [16]; [17].The tests were done with multiple payload sizes from 10 MB to 5GB files and OPDL improvement could be seen as the payload size increases where a 1GB file in application layer method took around 62 seconds in WiFi6 based device but OPDL only took 6.44 seconds on the same device which proves that the application layer tools faces a heavy delay due to the network overhead and memory handling [18]. The OPDL takes less than 2 seconds to discover multiple peers as it uses BLE and Wi-Fi action frames together [19] . Once a peer is found it does a secure handshake using AES-256-GCM encryption that allows devices to connect quickly and securely. However there are some drawbacks where unlike photon or other application layer tool OPDL needs root access and specific hardware access to operate which makes cross-platform interoperability difficult for the normal users to install and use it [5] .In addition due to continuous scanning and rotating security key the cpu usage will increase 2 to 4 % depending upon the processor [8]; [20]. .

One of the Major limitation of the current OPDL is that it lacks a proper GUI interface where it mainly operates on the command line and a basic debug ui which is not suitable for the general users and another gap is that the OPDL is built for limited hardware and certain kernels as testing was done only on selected chipsets. The future work of OPDL will be focused on improving the GUI and making it compatable to multiple android devices and versions . The algorithm for finding peers can be optimized to limit the cpu usage of OPDL when it is idle. Need of Bluetooth Low Energy (BLE) in the streaming phase to reduce the battery consumption. With these improvements, OPDL can become a very useful tool for fast and safe sharing between devices

VI. CONCLUSION

The OPDL was to create devices that could discover and transfer data much faster than standard application level methods and internet solutions. The existing solutions are slow because the data passes through multiple software layers, which add delays to the transfer. To solve this, a system called OPDL (Open Peer Discovery Layer), which operates at the kernel level and uses zero-copy algorithms, resulting in better speeds and performance. In testing, finds that the old method took around 62 seconds to transfer a 1 GB file, whereas OPDL took approximately 6.4 seconds , shows that OPDL is 10.9 times faster than the traditional approach, Hence OPDL Makes device to device communication much faster, efficient, secure through AES-256-GCM encryption and supports cross-platform interoperability.

REFERENCES

[1] Hegde, Abhilash Shreedhar, Amruta Narayana Hegde, Adeep Krishna Keelar, and Ananya Mathur. 2023. "Photon: A Cross Platform P2P Data Transfer Application." *arXiv [Cs.NI]*. https://doi.org/10.48550/ARXIV.2306.17173.

[2] Camps-Mur, D., A. Garcia-Saavedra, and P. Serrano. 2013. "Device-to-Device Communications with Wi-Fi Direct: Overview and Experimentation." *IEEE Wireless Communications* 20 (3): 96–104.

[3] Chen, Shiyan, and Dagang Li. 2018. "Efficient Zero-Copy Mechanism for Intelligent Video Surveillance Networks." *MATEC Web of Conferences* 189: 03021.

[4] Lee, Myungsup, Taeseop Lee, Hyung-Sin Kim, and Saewoong Bahk. 2022. "Wi-BLE: On Cooperative Operation of Wi-Fi and Bluetooth Low Energy under IPv6." *Journal of Communications and Networks* 24 (4): 489–502.

[5] Stute, Milan, David Kreitschmann, and Matthias Hollick. 2018. "One Billion Apples' Secret Sauce: Recipe for the Apple Wireless Direct Link Ad Hoc Protocol." In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM. https://doi.org/10.1145/3241539.3241566.

[6] Mei, Benshan, Wenhao Wang, and Dongdai Lin. 2025. "SVSM-KMS: Safeguarding Keys for Cloud Services with Encrypted Virtualization." In *Lecture Notes in Computer Science*, 217–35. Lecture Notes in Computer Science. Singapore: Springer Nature Singapore.

[7] Shen, Wenlong, Bo Yin, Xianghui Cao, Lin X. Cai, and Yu Cheng. 2016. "Secure Device-to-Device Communications over WiFi Direct." *IEEE Network* 30 (5): 4–9.

[8] T'Jonck, Kristof, Bozheng Pang, Hans Hallez, and Jeroen Boydens. 2021. "Optimizing the Bluetooth Low-Energy Service Discovery Process." *Sensors (Basel, Switzerland)* 21 (11): 3812.

[9] Barua, Arup, Md Abdullah Al Alamin, Md Shohrab Hossain, and Ekram Hossain. 2022. “Security and Privacy Threats for Bluetooth Low Energy in IoT and Wearable Devices: A Comprehensive Survey.” *IEEE Open Journal of the Communications Society* 3: 251–81.

[10] Dai, Yifan, Jacopo Tagliabue, Andrea Arpaci-Dusseau, Remzi Arpaci-Dusseau, and Tyler R. Caraza-Harter. 2025. “Zerrow: True Zero-Copy Arrow Pipelines in Bauplan.” *arXiv [Cs.OS]*. https://doi.org/10.48550/ARXIV.2504.06151.

[11] Al-Omoush, Khaled Saleh, Daniel Palacios-Marqué, Samuel Ribeiro-Navarrete, and Klaus Ulrich. 2024. “The Innovative Impact of Trust in Social Government: An Analysis during COVID-19 Crisis.” *IEEE Transactions on Engineering Management* 71: 549–59.

[12] Ranjan, Rajiv, Aaron Harwood, and Rajkumar Buyya. 2008. “Peer-to-Peer-Based Resource Discovery in Global Grids: A Tutorial.” *IEEE Communications Surveys & Tutorials* 10 (2): 6–33.

[13] Datta, Soumya Kanti, Christian Bonnet, and Navid Nikaein. 2014. “Self-Adaptive Battery and Context Aware Mobile Application Development.” In *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE. https://doi.org/10.1109/iwcmc.2014.6906452.

[14] Papadimitratos, Panagiotis, and Zygmunt J. Haas. 2024. “Secure Routing for Mobile Ad Hoc Networks.” *arXiv [Cs.CR]*. https://doi.org/10.48550/ARXIV.2403.00404.

[15] Hirofuchi, Takahiro, Hirotaka Ogawa, Hidemoto Nakada, Satoshi Itoh, and Satoshi Sekiguchi. 2009. “A Live Storage Migration Mechanism over WAN for Relocatable Virtual Machine Services on Clouds.” In *2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*. IEEE. https://doi.org/10.1109/ccgrid.2009.44.

[16] Suresh, J., A. Srinivasan, and A. Damodaram. 2010. “Performance Analysis of Various High Speed Data Transfer Protocols for Streaming Data in Long Fat Networks.” In *2010 International Conference on Recent Trends in Information, Telecommunication and Computing*. IEEE. https://doi.org/10.1109/itc.2010.64.

[17] Oğuz, Barlas, Venkat Anantharam, and Ilkka Norros. 2011. “Stable, Scalable, Decentralized P2P File Sharing with Non-Altruistic Peers.” *arXiv [Cs.NI]*. https://doi.org/10.48550/ARXIV.1107.3166.

[18] Høiland-Jørgensen, Toke, Michał Kazior, Dave Täht, Per Hurtig, and Anna Brunstrom. 2017. “Ending the Anomaly: Achieving Low Latency and Airtime Fairness in WiFi.” *arXiv [Cs.NI]*. arXiv. http://arxiv.org/abs/1703.00064.

[19] Zhang, L., C. Feng, and T. Xia. 2025. “Optimization of BLE Broadcast Mode in Offline Finding Network.” *arXiv [Cs.NI]*. https://doi.org/10.48550/ARXIV.2504.01422.

[20] Xiao, Yao, Sitian Chen, Amelie Chi Zhou, Shuhao Zhang, Yi Wang, Rui Mao, and Xuan Yang. 2024. “Low-Latency Video Conferencing via Optimized Packet Routing and Reordering.” In *2024 IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS)*, 1–10. IEEE.