International Journal of Basic and Applied Sciences, 14 (2) (2025) 93-99

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/9m31a384
Research paper

Real-Time Streaming Relay Device for P2P Overlay Networks In Pervasive Computing Environments

T. Padmapriya [1] *, Dr. Veeramani V [2], Dr. Antonyraj Martin [2], Nirmalkumar V [3], Gomathi C [4]

[1] Melange Publications, Puducherry, India
[2] Preparatory Studies Center, Mathematics & Computing Skills Unit, University of Technology and Applied Sciences, Salalah, Sultanate of Oman
[3] Assistant Professor, Artificial Intelligence and Machine Learning, St Joseph's College of Engineering (Autonomous), Tamilnadu.
[4] Assistant Professor, AI&DS, Panimalar Engineering College, Chennai
**Corresponding author E-mail: padmapriyaa85@ptuniv.edu.in*

Received: May 6, 2025, Accepted: May 18, 2025, Published: June 10, 2025

Abstract

P2P overlay networks have garnered much attention from researchers because they are useful as virtualized abstractions of intricate network architectures that can be tailored to meet requirements, the shortest diameter, or the least delay. In this paper, we discuss relevant studies on P2P (peer-to-peer) overlay networks in ubiquitous settings. Therefore, we examine pertinent needs and discuss the implementation and spread of P2P techniques and overlays on top of the networking infrastructures that are supported by ubiquitous environments. Novel techniques were developed to solve these difficult network problems with traditional P2P systems, assisting researchers in creating new application layer networks on the shelter of pre-existing P2P networks. In this paper, we contribute by analyzing the main ideas of next-level P2P (NL P2P) and methodically characterizing them. We hope to stimulate more study in this area by outlining and evaluating present systems, as well as by talking about ongoing studies and unresolved problems. At the same time, this work should be used as a benchmark for the most advanced P2P overlays in ubiquitous settings. Based on the application-layer multicast (ALM) structure and the hierarchical overlay networks, the real-time streaming relay mechanism may effectively increase the transportation efficiency of conference stream exchange.

Keywords: *Pervasive Computing; P2P (Peer-To-Peer); Application-Layer Multicast; Networking.*

1. Introduction

Pervasive computing environments have garnered a lot of research interest and have proven to be quite useful in business contexts by offering end users smooth, personalized, and inconspicuous services across heterogeneous infrastructures. Pervasive computing is the concept of providing value-added products and services that are user-centric and responsive to monitored conditions, such as context information, at any time and from any location.

Pervasive environments naturally encourage flexibility, availability, and adaptability as well as mobility because they are constructed on top of distributed, high-capacity networking systems [1]. The requirement to enable and improve widespread computing scenarios has led to a proliferation of wired, wireless, and mobile networking technologies in recent years. In this regard, the underlying networks ought to be adaptable and expandable enough to support this degree of individualization and diversity among users and applications operating on top of ubiquitous environments. Therefore, considering pertinent issues like dynamicity, scale, complexity, and heterogeneity, achieving the goal stated by pervasive computing comes at a significant cost.

In the sections that follow, we examine the state-of-the-art P2P overlays using a common classification system that distinguishes between structured and unstructured overlays. The purpose of the classification is to draw attention to the variations in the topological characteristics of each overlay. We also broaden our review to include bioinspired P2P overlays or those whose lookup and topology are modeled after biological and natural processes. Finally, we also peek at multi-layer overlays that consider several overlays working together in tandem and that take advantage of the known synergies between the latter. An overview of the examined systems using the chosen classification methodology is shown in Figure 1.1.

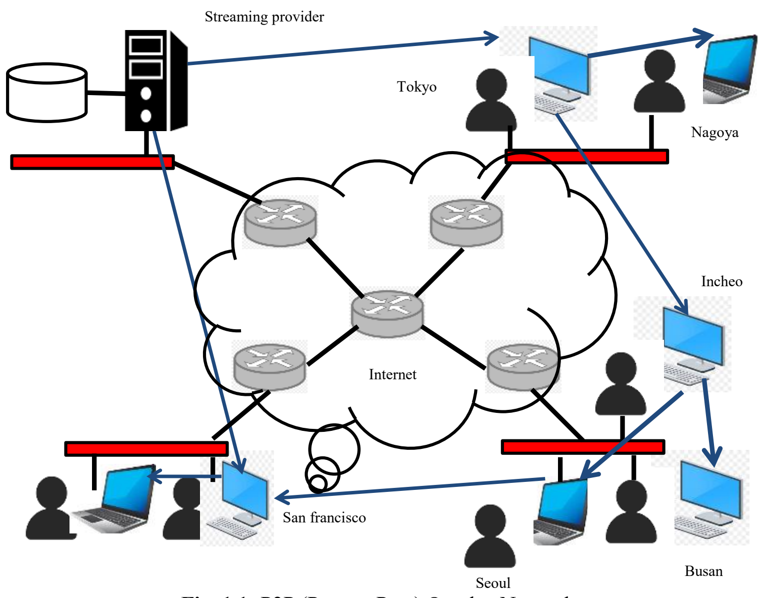

Fig. 1.1: P2P (Peer-to-Peer) Overlay Network.

A key paradigm for creating large-scale applications with characteristics of digital ecosystems is peer-to-peer (P2P) networks. P2P networks [2] are decentralized and capable of achieving good scalability. It is possible to divide the system's computing load among peer nodes in P2P networks. Thus, by communicating, assisting, and managing the systems' assets, users transform into their actors. P2P networks are highly intriguing for the creation of decentralized applications because of this feature. P2P networks have developed from straightforward file-sharing platforms for Internet users to game-changing tools for social and cooperative endeavors [3]. Indeed, in intelligent and interactive contexts, these systems can provide regular users with happy assessment, arrangement, and management. Consequently, P2P technologies provide the framework for creating apps that assist any group of individuals with shared political, cultural, scientific, and technological interests.
In Section 2, we go over the characteristics of different P2P simulators in detail. Section 3 presents and categorizes different P2P network emulators based on their attributes, and Section 4 discusses them. Finally, we address the findings of the case study and the conclusion of the P2P simulator survey in Section 5.

2. Related works

It was the initial system to realize that numerous peers with the required content could handle requests for common material rather than sending them to a central server. These P2P file-sharing networks are self-scaling, meaning that their total download capacity increases as additional peers sign up [4]. Napster uses an administrative search facility that is based on file lists supplied by each peer to achieve this self-scaling behavior; as a result, the centralized search does not require a lot of throughput. Because of the centralized search method, such a system has the problem of having a single point of failure. But Napster's killer app, the free-for-all file-sharing program for unauthorized digital music, was the target of a lawsuit brought by the Recording Industry Association of America (RIAA) that sought to shut it down.
The latest generation of mobile communication networks (3G) is gaining traction and beginning to be utilized globally despite several technical issues and the economic slump. People can now roam and chat, or have constant access to a communication network no matter where they are, thanks to the second version of these systems (2G). Mobility has entered our lives thanks to the capacity to roam and chat, which has also sparked the creation of several mobile apps [5]. Mobility, such as high-speed access to the web, media, knowledge, and e-commerce services wherever we are, not only at our desktop computers, home PCs, or screens, is now seen as a basic component of every application rather than an add-on.
P2P over ZeroNet, which makes use of ZeroNet, was developed primarily to address security concerns associated with I2P and Tor. The new secure method makes use of the two-layer P2P protocol. As a result, the peers are gradually regenerated. The anonymity features of such a system are what make it advantageous. Because these systems use two-layer P2P protocols, they also thwart tracking and debugging. Two P2P layers make use of Tor's onion router [6] to select superhubs and ZeroNet's site architecture. P2P over ZeroNet systems consists of three main modules: the secure transfer module, the distant receiving module, and the local editing module. User setups, packet thinking, and other operations are done in the local module. The local editor module also conveys packet encryption.
There are two types of traditional methods for finding services in ubiquitous environments: dispersed protocols and centralized protocols. They require a centralized lookup server to function. It is possible to further divide the current distributed protocols into two categories: organized and unstructured [7]. Unstructured searches are blind, but structured searches are routed. Distributed protocols seem to be more appropriate in a mobile, pervasive context than centralized protocols. Distributed protocols may create excessive overhead and need a lot of energy to sustain network layouts, making them unsuitable for pervasive environments. A service hierarchy is used in GSD to categorize services into various classes. The service descriptions are timed, televised advertisements. This advertising strategy uses flooding to drive a lot of traffic.
The fact that p2p networks are usually founded on completely different assumptions than those held in ad hoc networks is one of the primary barriers to this integration. The capacity of legacy peer-to-peer systems is millions of nodes. Additionally, they are designed for a networking environment that is usually resource-rich, particularly in bandwidth [8]. As a result, legacy P2P systems typically sacrifice bandwidth in favor of more flexibility. Flat multi-hop ad hoc networks do not have significant scalability issues up to hundreds of locations. Actually, due to inherent wireless capacity limitations, both theoretical and practical data indicate that large-scale flat ad hoc systems are unlikely. Based on these results, an "ad hoc horizon" is defined for actual flat dynamic networks with two to three hops between peer-to-peer connections and 10 to 20 nodes.

The development of pervasive technology encourages the creation of sophisticated apps with context awareness enabled. There have been numerous attempts to create different context-aware systems throughout the last ten years. How to extract high-level context information from a collection of low-level context information that might be dispersed across different fields is one of the difficulties with these systems [9]. A centralized method to context reasoning has been implemented by numerous systems in recent years. This strategy is effective in a single smart area, such as a smart home. Due to the modest size of the info set in a single domain, it can respond quickly and update background information or indexes with relative ease.
Individual peers in realistic peer-to-peer (P2P) systems are frequently very unreliable, offline more often than online, and may even be malevolent or self-serving. Therefore, replication is required to achieve resilience and dependability requirements. But updating the duplicates is a necessary step in the replication process. Because peers store anything, it is impossible to identify replicas in unstructured P2P systems. Fortunately, structured P2P systems have a defined subnetwork that replicates a specific articular key space. Peers' physical addresses will frequently change due to the limited number of IPv4 addresses and regular interruptions and reconnections. Handling peer identification is crucial as a direct result of this [10]. Peers in all structured DHT-based P2P systems specialize in a certain key space; hence, it is essential to identify them for proper routing, even though in uncontrolled P2P platforms, this may not be very relevant because flooding must be utilized nevertheless.

3. Methods and materials

3.1. Properties of P2P simulators

The following criteria can be used to compare the P2P simulators and choose the one most appropriate for application testing and implementation:

3.1.1. Simulator architecture

The architecture outlines the fundamental features, implementation specifics, and characteristics of the simulator's design and operation. The following are several perspectives for examining the simulator architecture:

- P2P structure: It shows if structured overlays, unstructured overlays, or both are supported by the P2P simulator. Peers in structured P2P overlay networks are arranged by a predetermined topology, and algorithmic decisions are made to preserve the network's topology and characteristics. Usually, they employ tree-based indexing or Distribution Hash Tables (DHTs) indexing [11]. There is no algorithm for organizing or optimizing network connections in unstructured peer-to-peer networks. It consists of peers who join the network at random and don't know the topology beforehand. In a configuration file that is readable by humans, the user needs to be able to set all pertinent simulation parameters. Additionally, the simulator should support node failure, malicious node behavior, and dynamic node behavior.
- Simulator nodes: The simulator mode shows whether cycle-based simulation, discrete event simulation, or both are supported by the simulator. A system's activity is depicted as a series of events in discrete event simulation. It makes use of a scheduler that adds delay if needed and synchronizes message transfers across nodes. Every event signifies a shift in a system's state and happens at a specific moment in time. Cycle-based simulator is a time-driven consecutive simulation in which every node in each cycle performs the activities of each protocol sequentially. The event-based engine can also run cycle-based protocols, but not the other way around.
- Underlying network simulation: P2P simulators use a variety of techniques to model the simulation layer or the underlying network. They fall into one of two categories: flow-based or packet-based. For each packet created or used by simulations, packet-based simulators, like NS-2, compute delay, bandwidth, and routing. They also mimic the packet and links. Other simulations are flow-based and abstract away the features below the simulator layer, typically without mapping to the network underneath or even considering its layout. They operate at the program layer and facilitate communication between connected peers via protocols for transportation like TCP or UDP.
- Underlying protocol simulation: The way that P2P simulators relate to the fundamental protocols varies. They fall into one of three categories: domain-specific, protocol-specific, or generic simulators. Any P2P application can be simulated using generic simulators. Domain-specific simulators replicate P2P systems in a particular field, while protocol-specific simulators are made to replicate a particular protocol.
- The support of the distributed parallel simulation: Compared to when it is done sequentially, the task will execute more quickly if it is distributed in parallel across multiple machines. For distributed situations, it is crucial to take into account if the P2P simulator allows exercises to be run over multiple machines. Higher scalability or a quicker simulation runtime are made possible by this. Parallel execution, however, necessitates verifying the dependence controls when distributing over various machines.
- The support of churn: The churn rate, or the rate at which nodes join or depart the network per unit of time, is used to model how a peer or node might behave in a dynamic environment [12]. These properties also include whether the simulation records node additions and deletions, what churn levels are emulated, and whether node failure (either temporary or permanent) is handled.

3.1.2. Usability

The simulator's ease of use and learning are gauged by its usability. The simulator's documentation should be thorough, precisely specified [13], expandable, and simple to read for this reason. Furthermore, end users should find the testing and experimentation settings convenient. The simulator documents' script languages or interfaces should be descriptive and simple to understand.

3.1.3. Scalability

The simulator's scalability is determined by the practical network size that it can model, typically expressed in the form of the total amount of nodes. Scalability is a crucial and difficult feature for confirming a simulator's performance because P2P applications typically have rather large networks. Therefore, a simulator that offers more scalability—that is, thousands of nodes or more—helps carry out continuous tests that are challenging to execute on hundreds of machines in real-world situations. In addition, making effective use of the computing resources at hand is a crucial component in enhancing scalability.

3.1.4. Statistics

The outcomes that a simulator generates and how they are saved for later use are also crucial components. For statistical analysis to be performed, the results must be easily manipulable, internationally expressive, and related to engineering science and technology. In order to verify the reproducibility of results, mechanisms that enable experiment repetition should be in place, such as the capacity to save simulator states.

3.1.5. Interactive visualizer

It should be possible to test and debug new or current overlay protocols using a visualizer, like a graphical user interface (GUI). The visualizer displays the overlay topology and the underlying network architecture in a personalized manner. In addition to the above-mentioned requirements, we take into consideration the following additional criteria for various simulators because our research additionally concentrates on the execution of the P2P protocols through a case study:

3.1.6. P2P protocols implemented

The fundamental protocols that the P2P simulator implements are included in this attribute. Based on the fundamental protocols used that are comparable to the needed application, a simulator can be chosen. It is possible to extend or reuse the built-in overlay protocol implementations for actual network applications. These characteristics allow for a comparison of the current simulators. Nevertheless, not every simulator adheres to every one of the properties that have been presented. In the following section, we go over a variety of simulators and their characteristics.

3.2. Various P2P network simulators

We evaluate P2P simulators based on their applicability to different protocols, including generic, protocol-specific, and domain-specific simulators for specific fields.

3.2.1. Generic simulators

The generic P2P simulator includes commonly used modules for protocol simulations, allowing users to focus on the core aspects of the network or application. The simulator's clear structure and modular construction allow for easy extension and replacement of protocols and applications. Several generic simulators exist, including PeerSim, OverSim, 3LS, and PlanetSim.

3.2.2. PeerfactSim.KOM

PeerfactSim.KOM is a Java-based simulation built for large-scale peer-to-peer applications. The simulation begins with an XML-based preparation file that specifies the layers.

3.2.3. Simulator architecture

PeerfactSim.KOM is a general discrete event simulator that supports both organized and unstructured overlays. The simulator is designed with a tiered architecture that includes an application layer, a service layer, an overlay layer, a transport layer, and a network layer to cover all aspects of a P2P network. Each layer contains several interfaces that provide functionality to the levels below it. It takes the network layer into account when communicating; hence, it fully supports message-based packet-level transmissions. PeerfactSim.KOM offers a churn generation built around a mathematical algorithm that handles node joins and departures.

- Usability: PeerfactSim.KOM's website provides rich documentation. Each layer has interfaces that connect to other layers and deliver services. Based on these interactions, the simulator introduces the concept of basic and skeleton implementations.
- Statistics: PeerfactSim.KOM includes its infrastructure for collecting data from active simulations. It employs a logging structure to trace and troubleshoot a simulation, as well as a statistics structure to extract relevant data for on-the-fly analytics or later post-processing.
- Interactive visualizer: The embedded visualization component visualizes the simulated P2P system's architecture as well as the exchanged messages. In terms of topology presentation, the visualization can arrange the peers depending on the network layer's coordinates or in a ring-like structure.
- P2P Protocols implemented: Unorganized covers: GIA, Gnutella 0.4, Gnutella 0.6, Napster; Structure overlays: CAN, Chord, CDHT, Kademlia, Pastry, and Globase.

3.3. Multi-layer P2P connections

The earlier description of the several P2P overlays that are currently in use, both structured and unstructured, emphasized both their unique advantages and disadvantages in terms of performance and operating context. There isn't a single solution that can be applied to all the problems associated with pervasive networks when it comes to P2P overlays.

In this regard, it is reasonable to expect that several overlays will be running concurrently on top of the same physical network infrastructures. This expectation is supported by existing network implementations. Another explanation for this is that most P2P overlays are domain- or application-specific, such as file sharing or VoIP. Since it is normal for nodes to be active in multiple domains or activities these days, the need for multiple overlays to coexist becomes more apparent. A typical setup of several P2P overlays operating concurrently on top of the same networking equipment is shown in Figure 3.1 [14]. P2P overlays 1 through N function simultaneously over the same physical network, as shown in Figure 3.1's lower section. This suggests that these P2P overlays' topologies are simultaneously controlled and consider both the efficiency standards established by the P2P overlay design and modifications in the actual networks. The ability of the coexisting P2P coverings to interact with one another depends on the algorithm used to create and uphold the multi-layer P2P overlay.

Furthermore, the inherent virtualization benefit of numerous overlays drives the need for them.

According to P2P overlays, network virtualization allows numerous applications to use the same physical space. Simulation research, however, could not always produce reliable findings. Since they should only serve as a supplement to actual trials because they contain numerous assumptions about reality deployments. Platforms like PlanetLab and VINI are common examples of how network virtualization using P2P overlay networks may significantly aid in the creation of innovative network protocols and their implementation in actual network environments.

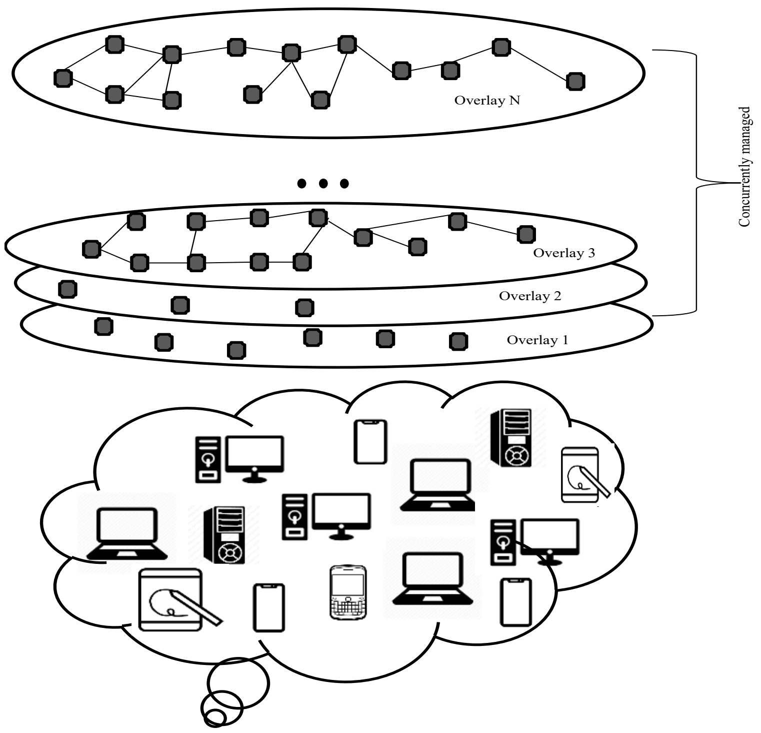

Fig. 3.1: Multiple Overlays Are Often Deployed on Top of the Same Physical Network.

P2P overlays enable many testbed experiments to occur simultaneously, over the same Internet connection, and under realistic settings, meeting the increasing demand for "live" trials for novel Internet infrastructure and procedures. Because of economies of scale and the elimination of potential runtime faults that computational studies cannot predict, such an approach results in notable cost reductions.

4. Implementation and experimental results

The suggested design of the P2P computing system was examined using the simulator mentioned in the preceding section.

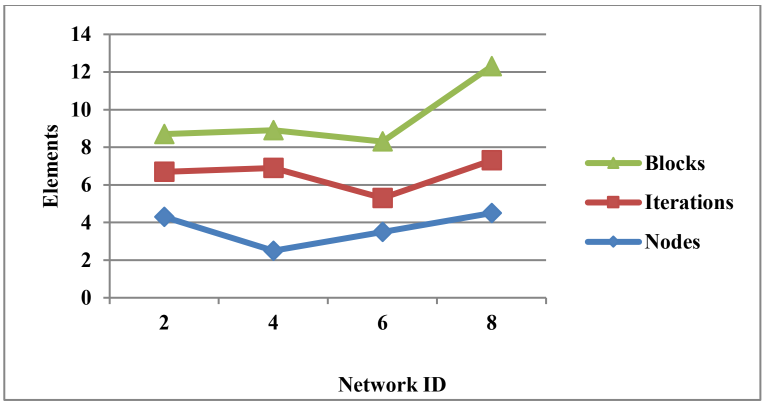

Fig. 4.1: Tested Networks' Parameters.

The OPEX cost of the system, which includes transfer and computation expenses, is the performance metric shown in the tests. According to the number of nodes, blocks, iterations, and other characteristics, we constructed ten systems (Figure 4.1). In accordance with the parameters of actual overlay systems, additional parameters (processing power, transfer cost, processing cost, and access link capacity) were randomly generated.

The results indicate that because unicast is not highly flexible in terms of data transfer, the employment of different policies has little effect on the cost for unicast flow.

Furthermore, there is a starving effect when the Cheapest-Available [15], Cheapest-Owner, and Rarest-Missing policies are used simultaneously in the case of the anycast flow and the Rarest-Missing method in the case of the unicast transmission flow (certain nodes are not able to obtain all result blocks in a given period). The starving effect has no influence whatsoever on the P2P flow. Furthermore, using policies rather than other methods greatly lowers the cost in the P2P flow scenario, resulting in a higher-quality solution.

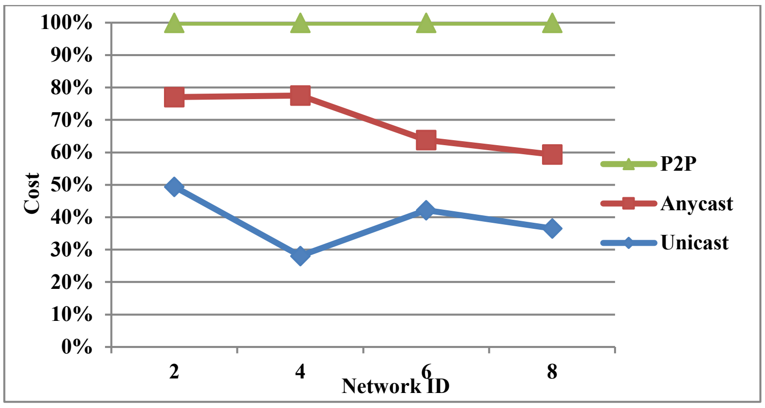

Fig. 4.2: An Expense A Business OPEX Cost concerning Different Kinds of Flows.

Comparing the OPEX costs for the three network flow types used for data distribution—unicast, anycast, and P2P—was the next objective of the simulations. The equivalent outcomes for the same ten systems are shown in Figure 4.2. It is easy to see that the P2P computing system works noticeably better than the unicast and anycast systems; on average, the cost reductions are 70% and 55%, correspondingly. Furthermore, the unicast flow is the one that is most affected by the increasing number of nodes, blocks, and other factors that define the problem's magnitude. This results from the fact that P2P and anycast flows are more flexible than unicast flows. For all tested networks, the Peer-to-Peer flow that has no further limitations on the sending node selection is the most resilient to the growth of problems and can maintain a slight cost increase. While the unicast communications flow consistently produces the worst solution in terms of cost, the P2P technique offers the best option for every investigation.
In addition, we experimented to compare our distributed system to conventional centralized computing platforms.
We built a computer system based on the BOINC architecture for the simulator system, which sends data to all participating nodes and collects results from one main server. Two scenarios of the server location were examined:

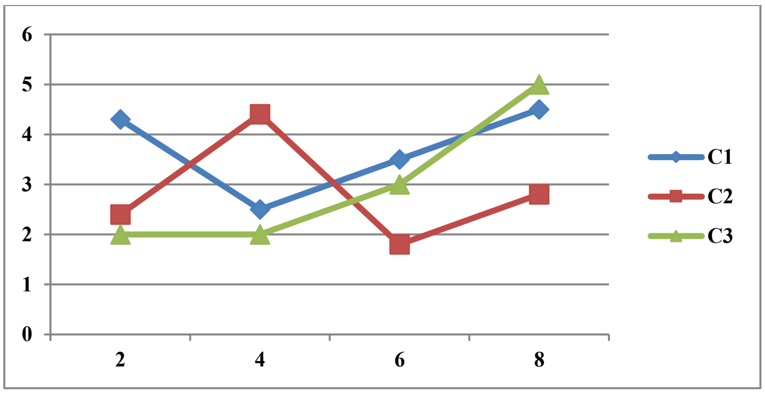

Fig. 4.3: The OPEX Cost for Centralized Systems without Server Location Optimization (C1), Dispersed Systems with P2P Flows (P2P), and Centralized Systems with Server Location Optimization (C2).

i) The distance to each node is the mean of all other distances, which is known as C1, indicating that the server location is not optimum;
ii) The server is positioned to minimize the distance to other nodes (designated as C2) in the optimal network location.

The findings in Figure 4.3 demonstrate that, in comparison to centralized scenarios C1 and C2, the distributed system significantly lowers costs; on average, the reductions are 70% and 66%, respectively. This experiment demonstrates that our distributed design can yield significant OPEX cost savings when compared to the centralized approach.
The second goal of the study was to examine the effects of adding new nodes to the system while keeping the amount of processed data constant. All network flows in this scenario experience an increase in cost (the same policy set was utilized as in the prior experiment). Both computation expenses and transfer fees (and thus the overall cost) satisfy this relationship. The process of joining additional nodes was repeated, and the simulation was run for every "new" network—that is, the network that had previously been joined with a new node. Every experiment reaches a point where no workable solution can be found since there isn't enough data to meet the fairness criteria, and the network capacity isn't large enough to allocate results to every node. Every network flow has a particular number of nodes at which the architecture stops providing a suitable answer; for example, a peer-to-peer flow always ends at the most nodes, a transmit flow always ends first, and an anycast flow always ends in the center.
Similar outcomes are seen in a different experiment when we raise the network's data volume while maintaining the same levels of computing power, network bandwidth, and other variables. Most blocks are always handled via the P2P method. Consequently, the P2P flow is the least expensive and immune to unwanted occurrences like node depletion. Furthermore, when adding new nodes to the network and expanding the volume of data in the framework, the P2P technique offers the highest performance.

5. Conclusion

Peer-to-peer security issues and several state-of-the-art solutions have been covered, along with the applicability and limitations of various schemes. We examined security while considering the attackers' goals, constraints, motivations, and points of attack. It appears that current security solutions are not developed enough to be implemented in peer-to-peer networks. The most difficult issues appear to be entity identification, association, and secure ID assignment.

Three types of network flows—unicast, peer-to-peer, and anycast—can be used by the system to distribute data. The technical specifications of the created real-time simulation system, as well as the suggested decision policies and system organization, have all been covered.

Discrete realistic simulation is a very useful method for optimizing networks. It frequently gets around the drawbacks of static optimization. In contrast to other network training simulators, ours allows you to simulate not just network traffic but also the shared computing process. We have concentrated on the environment, including input and output data, the simulator itself, and the potential for making the entire system adaptable and efficient.

In subsequent research, we suggest expanding the computation system with additional limitations, such as varying the number of nodes and replicated statuses throughout the whole simulation. The CDSim simulator can be used to assess new decision policies, which offer another avenue for future development.

References

[1] Malatras, A. (2015). State-of-the-art survey on P2P overlay networks in pervasive computing environments. Journal of Network and Computer Applications, 55, 1-23. https://doi.org/10.1016/j.jnca.2015.04.014.

[2] Barolli, L., & Xhafa, F. (2010). Jxta-overlay: A p2p platform for distributed, collaborative, and ubiquitous computing. IEEE Transactions on Industrial Electronics, 58(6), 2163-2172. https://doi.org/10.1109/TIE.2010.2050751.

[3] Liu, Y., Xiao, L., & Ni, L. (2007). Building a scalable bipartite P2P overlay network. IEEE Transactions on Parallel and Distributed Systems, 18(9), 1296-1306. https://doi.org/10.1109/TPDS.2007.1059.

[4] Lua, E. K., Crowcroft, J., Pias, M., Sharma, R., & Lim, S. (2005). A survey and comparison of peer-to-peer overlay network schemes. IEEE Communications Surveys & Tutorials, 7(2), 72-93. https://doi.org/10.1109/COMST.2005.1610546.

[5] Krčo, S., Cleary, D., & Parker, D. (2005). Enabling ubiquitous sensor networking over mobile networks through peer-to-peer overlay networking. Computer communications, 28(13), 1586-1601. https://doi.org/10.1016/j.comcom.2004.12.043.

[6] Naik, A. R., & Keshavamurthy, B. N. (2020). Next level peer-to-peer overlay networks under high churns: a survey. Peer-to-Peer Networking and Applications, 13(3), 905-931. https://doi.org/10.1007/s12083-019-00839-8.

[7] Yu, C., Yao, D., Li, X., Zhang, Y., Yang, L. T., Xiong, N., & Jin, H. (2013). Location-aware private service discovery in pervasive computing environment. Information Sciences, 230, 78-93. https://doi.org/10.1016/j.ins.2012.08.010.

[8] Delmastro, F., Passarella, A., & Conti, M. (2008). P2p multicast for pervasive ad hoc networks. Pervasive and Mobile Computing, 4(1), 62-91. https://doi.org/10.1016/j.pmcj.2007.03.001.

[9] Gu, T., Pung, H. K., & Zhang, D. (2008, March). Peer-to-peer context reasoning in pervasive computing environments. In 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom) (pp. 406-411). IEEE. https://doi.org/10.1109/PERCOM.2008.37.

[10] Datta, A. (2003, June). Mobigrid: Peer-to-peer overlay and mobile ad-hoc network rendezvous-a data management perspective. In Proc. of the CAiSE 2003 Doctoral Symposium.

[11] Peltotalo, J., Harju, J., Saukko, M., Vaatamoinen, L., Bouazizi, I., Curcio, I. D., & van Gassel, J. (2009, April). A real-time peer-to-peer streaming system for mobile networking environment. In IEEE INFOCOM Workshops 2009 (pp. 1-7). IEEE. https://doi.org/10.1109/INFCOMW.2009.5072102.

[12] Su, H. K., Pan, J. T., Chen, K. J., Ogiela, M. R., & Chen, H. C. (2019). Real-time streaming relay mechanism for P2P conferences on hierarchical overlay networks. Journal of Electronic Science and Technology, 17(3), 242-251.

[13] Chopra, D., Schulzrinne, H., Marocco, E., & Ivov, E. (2009). Peer-to-peer overlays for real-time communication: security issues and solutions. IEEE Communications Surveys & Tutorials, 11(1), 4-12. https://doi.org/10.1109/SURV.2009.090102.

[14] Hu, C. C. (2020). P2P Data dissemination for real-time streaming using load-balanced clustering infrastructure in MANETs with large-scale stable hosts. IEEE Systems Journal, 15(2), 2492-2503. https://doi.org/10.1109/JSYST.2020.2992774.

[15] Ramzan, N., Park, H., & Izquierdo, E. (2012). Video streaming over P2P networks: Challenges and opportunities. Signal Processing: Image Communication, 27(5), 401-411. https://doi.org/10.1016/j.image.2012.02.004.