International Science and Technology Journal
المجلة الدولية للعلوم والتقنية

Volume 36 العدد
Part 1 المجلد

http://www.doi.org/10.62341/mmle0834

Received	**2025/02/15**	**تم استلام الورقة العلمية في**
Accepted	**2025/03/03**	**تم قبول الورقة العلمية في**
Published	**2025/03/10**	**تم نشر الورقة العلمية في**

Evaluating the performance impact of routed protocols on the response speed of distributed networks

Moktar M. Lahrashe

Al-Zawiya College of Computer Technology- Libya
mlahrashe@gmail.com

Abstract

From the last two decades, the trends in the computing industry are towards distributed networks, high performance, and efficiency, which are becoming more and more important in the data-driven world of today. The need for high-performance, increasingly scalable networks makes it crucial to comprehend how various routing protocols affect response time. This study investigates how various protocols impact the reaction speed of dispersed networks. The scalability and effectiveness of dispersed networks are greatly influenced by routed protocols, which make it easier for data to be forwarded across various network portions. The study investigates how routing strategies, including link-state and distance-vector protocols, affect reaction time, throughput, and latency in a dispersed network setting by examining different routing schemes and network topologies. The study highlights the trade-offs between reaction speed and protocol complexity and identifies important aspects affecting distributed systems performance. According to the results, more complex routing protocols may result in overheads that impact response times even while they can improve routing efficiency with implications for bettering the design and administration of large-scale, data-intensive systems. This study offers insights on how to optimize routed protocols for improved performance in dispersed networks.

Keywords: Routed protocols, response Reaction time; distributed systems, response speed

International Science and Technology Journal
المجلة الدولية للعلوم والتقنية
Volume 36 العدد
Part 1 المجلد
http://www.doi.org/10.62341/mmle0834

تقييم تأثير الأداء للبروتوكولات الموجه على سرعة استجابة الشبكات الموزعة

مختار محمد الاحرش
كلية تقنية الحاسوب الزاوية – ليبيا
mlahrashe@gmail.com

الخلاصة

منذ العقدين الأخيرين، تتجه الاتجاهات في صناعة الحوسبة نحو الشبكات الموزعة والأداء العالي والكفاءة، التي أصبحت أكثر أهمية في العالم اليوم القائم على البيانات. إن الحاجة إلى شبكات قابلة للتطوير وعالية الأداء، تجعل من الضروري فهم كيفية تأثير بروتوكولات التوجيه المختلفة على وقت الاستجابة. تبحث هذه الدراسة في كيفية تأثير البروتوكولات المختلفة على سرعة تفاعل الشبكات الموزعة (Networks Dispersed). تتأثر قابلية التوسع وفعالية الشبكات المشتتة بشكل كبير بالبروتوكولات الموجه، مما يجعل من السهل على البيانات إعادة توجيهها عبر أجزاء شبكات المختلفة. تبحث هذه الدراسة في كيفية تأثير استراتيجيات التوجيه، بما في ذلك دور الارتباط وبروتوكولات المخرج عن بعد، على وقت التفاعل والإنتاجية (Throughput) والتاخير (Latency) في إعداد الشبكة المشتتة من خلال فحص مخططات التوجيه المختلفة وطبولوجيا الشبكات. تبرز الدراسة المفاضة بين سرعة التفاعل وتعقيد البروتوكول وتحدد الجوانب المهمة التي تؤثر في أداء الأنظمة الموزعة. وفقا للنتائج، قد تؤدي بروتوكولات التوجيه الأكثر تعقيدا إلى النفقات العامة التي تؤثر في أوقات الاستجابة حتى في حين أنها يمكن أن تحسن كفاءة التوجيه مع الآثار المترتبة على تحسين تصميم وإدارة أنظمة واسعة النطاق كثيفة البيانات. تقدم هذه الدراسة رؤى في كيفية تحسين البروتوكولات الموجه لتحسين الأداء في الشبكات الموزعة.

الكلمات الرئيسية: بروتوكولات التوجيه، زمن رد فعل الاستجابة، الأنظمة الموزعة، سرعة الاستجابة

1. Introduction

The response speed of distributed networks is a critical factor in ensuring efficient communication and seamless data transfer. The key to controlling this communication is the use of routed protocols, which control how information is sent across a network by directing data packets across various segments or subnets. These protocols allow the network to effectively deliver data packets to their

http://www.doi.org/10.62341/mmle0834

intended locations. however, because it directly impacts elements like latency, network congestion, and routing decision-making, the routed protocol selection can have a big impact on response time, to manage the complexity of data flow across several systems, distributed networks, which are made up of numerous interconnected nodes, need scalable and reliable routing methods. The performance of the network might vary significantly depending on the type of routing protocol utilized, such as distance-vector, link-state, or hybrid protocols. These protocols control the forwarding choices and routing information sent by network devices (routers), which in turn affects the overall speed and effectiveness of data transfer. The goal of this study is to investigate how various routing protocols affect dispersed networks response times. The goal of this study is also to provide light on how to optimize routing protocols for faster response times by examining different routing techniques and how they affect network performance. The results will improve the performance of contemporary, large-scale networks by advancing our understanding of how to balance routing efficiency with the least amount of latency in intricate distributed systems.

The study aims to comprehend how various routed protocols impact response times and efficiency in distributed network settings. Optimizing the performance of dispersed networks is crucial as they become more complicated and essential to many industries, from cloud computing to workplace networks. Although routed protocols are essential for controlling data transfer between various nodes and subnets, there is still much worry about how they affect response times.

The routing strategy chosen in dispersed networks, where data must travel via several nodes and communication channels, can slow down response times by increasing latency, decreasing throughput, or creating network congestion. The difficulty is determining how various routing protocols, such as link-state, distance-vector, or hybrid protocols, perform in various network topologies and comprehending the trade-offs between speed and complexity.

Even while some routing protocols could be more effective in particular situations, their use in large-scale, data-intensive systems may result in overheads that outweigh the advantages. In dynamic and complicated distributed settings, the challenge is striking a balance between the necessity of maintaining quick reaction times and the requirement for efficient routing. In order to identify the best

http://www.doi.org/10.62341/mmle0834

practices for lowering latency and enhancing network performance, this study aims to examine the link between routed protocols and the response speed of dispersed networks.

1.1 Importance of study

This study is significant because it has the potential to improve dispersed networks' performance and efficiency, which are becoming more and more important in the data-driven world of today. The need for high-performance, increasing scalable networks, makes it crucial to comprehend how various routing protocols affect response time. Network designers and engineers will benefit from this research's insightful findings on the connection between routing protocols and network performance, which will help them optimize their systems for faster response times, lower latency, and better data transfer.

The study is especially important for large-scale distributed systems where fast and dependable data transfer is critical, such as cloud computing infrastructures, business networks, and data centers. The study will help choose the best routing techniques depending on network demands and settings by determining how different routing protocols (such as distance-vector, link-state, and hybrid) affect response speed.

The results of this study will advance the subject of network optimization by providing workable answers to typical problems, including data transfer delays, network congestion, and routing overhead. The ultimate goal of this study is to enhance dispersed networks performance and efficiency so that businesses may manage growing data volumes more effectively while preserving high standards of dependability and service quality.

2. Methodology

A general literature review was conducted to learn more about the comprehensive analysis of the body of research on routed protocols and how they affect the performance of dispersed networks as part of the methodology's first stage. Different routing protocol types (distance-vector, link-state, and hybrid protocols) and their effects on network latency, throughput, and overall efficiency will be the main topics of this review to identify research gaps and direct the experimental design. Important papers on network performance, protocol behavior under various settings, and optimization strategies will be reviewed.

http://www.doi.org/10.62341/mmle0834

For the routing protocol, the research will choose a number of widely used routing protocols, including EIGRP (Enhanced Interior Gateway Routing Protocol), OSPF (Open Shortest Path First), and RIP (Routing Information Protocol). These protocols will be selected according to their variations in distributed network performance, scalability, and complexity.

To assess how well each routing protocol works in various network configurations, the simulation will mimic a variety of distributed network topologies, such as star, mesh, and hybrid architectures. Control over variables like network load, node behavior, and routing updates, all of which are crucial for evaluating protocol performance. The Cisco Packet Tracer, GNS3, and Omnet are examples of network simulation technologies that can be used to establish a simulated network environment.

To determine how each routing protocol affects response time, the experimental design will test each one in comparable network scenarios. Important assessment metrics will consist of:

- **Latency**: The amount of time it takes for data packets to go from their source to their destination is known as latency.
- **Throughput**: The volume of information that is successfully sent via a network in a specific length of time.
- **Packet Loss**: The proportion of data packets lost during transmission as a result of ineffective routing or network congestion.
- **Jitter**: Variability in packet arrival timings that might impact programs that operate in real time to learn how protocols scale in response to rising data needs, the tests is carried out under various network loads.

The data collection and analysis show how each routing protocol affects the network's reaction speed. The data gathered from the network simulations is examined for the performance of the routing protocols in various network topologies and traffic situations using statistical analysis techniques including mean latency, standard deviation, and performance ratios.

3. Literature review

The literature survey first aims to provide an overview of previous works of routed protocols in dispersed networks. Then, in the background, we discuss the theory of distributed systems and directed network protocols and how response speed is affected by different routing strategies under varied network circumstances.

http://www.doi.org/10.62341/mmle0834

3.1 Research trends

There are several works that have been carried out by other researchers in this field. However, this section gives an overview of some important research papers, challenges, and problems of distributed systems toward protocol behavior under various settings and optimization strategies. Many authors have identified different issues of distributed systems. Authors in [1] study is to evaluate and compare how well different routing protocols perform in terms of load sharing, link failover, and overall network performance. The results of this research show that EIGRP has a better failover convergence time and packet loss percentage as compared to OSPF. The work of [2] described the issues in testing component-based distributed systems related to concurrency, scalability, heterogeneous platforms, and communication protocols. Authors in [3] assess the effectiveness of routing protocols within an IPv6 network, particularly RIPng, OSPFv3, and EIGRP. Performance measures such as throughput, jitter, and packet loss are evaluated using GNS3, with the goal of determining the suitability of each routing protocol in an IPv6 context. Authors in [4] compare the performance of an Interior Gateway Routing Protocol (IGRP) with an Exterior Gateway Protocol (EGP), aiming to identify the best protocol combination for complex environments. The study also simulates the Hot Standby Routing Protocol (HSRP) and the Gateway Load Balancing Protocol (GLBP) to evaluate load balancing and redundancy specific to BGP. Authors in [5] explored two eminent protocols, namely, Enhanced Interior Gateway Routing Protocols (EIGRP) and Open Shortest Path First (OSPF) protocols. The evaluation of these routing protocols was performed based on the quantitative metrics such as Convergence Time, Jitter, End-to-End delay, Throughput and Packet Loss through the simulated network models. The evaluation results showed that EIGRP routing protocols provide better performance than OSPF routing protocols for real-time applications. Authors in [6] analyze the result of the performance of various routing protocols, naming RIP, OSPF, IGRP, & EIGRP. Over transmission cost, router throughput, and delay. The study shows the OSPF has the finest act as a whole. Router overhead as RIP, EIGRP performs well as above OSPF. For the best service, OSPF transmits a packet better than others. The introduced work in [7-10] addressed that there are very few comparisons and analyses on all IGP protocols for real-time application that have been made. For the most part, previous studies

http://www.doi.org/10.62341/mmle0834

of different routing protocols such as EIGRP and OSPF have been done based on simulation [8], in which the authors have concentrated on comparative performance and a detailed simulation study carried out in the IP network. In the area of interior routing protocols, numerous studies have been published about the behavior of OSPF, RIP, IS-IS, and EIGRP [7-10]. These studies have contributed a lot of potential insights on interior routing protocols, which has drawn similar attention to work in that direction. Author in [11] focus on Massively Distributed Systems: Design Issues and Challenges. Authors in [12] focus on scheduling problems for a class of parallel distributed systems. Authors in [13] described the Distributed Computing: Principles. Author in [14] described distributed software engineering. Authors in [15] presented the Monitoring of Distributed Systems. Authors in [16] proposed a technique to improve fault tolerance and reliability against faulty or unreliable clouds and designed a model specifically to address the problem of selection of cloud environments to achieve better and more reliable results.

3.2 Background theory

3.2.1 Distributed Systems

Since the invention of the first computer devices, computing has undergone several changes. Fast and affordable processors are now available thanks to technological advancements, and profitable and highly skilled computer networks are now possible because of advancements in communication technologies. One of the components of centralized networks is constantly shared by users; although all resources are available, there is only one point of failure and one point of control. In the late 1970s, a new computing paradigm known as distributed computing emerged as a result of the fusion of computer and networking technology. Distributed computing has revolutionized computing by providing rapid and accurate answers for a wide range of challenging issues in several domains. We spend more time talking and obtaining information online these days as we are so consumed by the digital age. The Internet continues to advance by several orders of magnitude, enabling end systems to connect in ever-more-different ways. From basic data exchange to sophisticated systems that support a wide range of services, a number of techniques have developed throughout time to facilitate these advancements [17]. The biggest shift in information technology over the past 20 years has been the proliferation of networked workstations and the decline of the

http://www.doi.org/10.62341/mmle0834

centralized mainframe. This change has dispersed hardware resources and given end users access to more computing power. When computers collaborate via a network, the network may draw power from all of the connected computers to carry out intricate operations. One may categorize computation in networks of processing nodes as either distributed or centralized. In a centralized solution, the central system is shared by all nodes, and one node is assigned to handle the full application locally. The users consistently, as a result, have just one point of failure, and control of the availability of inexpensive, powerful computers and network technologies is what is driving the expansion of decentralized computing. The total amount of processing power that can be used can be astoundingly large when a few strong computers are connected and interact with one another. Compared to a single supercomputer, such a system may have a greater performance share. One potentially extremely effective method for gaining access to a lot of computer power is distributed computing, which is a decentralized approach to computing. These systems aim to reduce the cost of computing and communication. The application's processing stages are split up across the participating nodes in distributed systems; the fundamental stage in all the ideas of computer-to-computer communication is central to distributed computing systems [18].

An application that uses a set of protocols to coordinate the activities of several processes on a communication network so that all of the parts work together to complete a single job or a limited number of related tasks is known as a distributed system the through the communication network, cooperating computers may access both local and remote resources in the distributed system in a distributed system, the user is aware that there are several independent computers the user is unaware that the tasks are carried out by several computers located in different places. This implies that, similar to centralized systems, no single machine inside the system is responsible for using all of the system resources needed to perform a computer program [19].

Architecture of Distributed Systems

Operating systems and networking software already in place provide the foundation for distributed systems. A distributed system is made up of several independent computers connected by distribution middleware and a computer network to become independent there two computers on the network have a distinct

http://www.doi.org/10.62341/mmle0834

master/slave relationship by allowing computers to share resources and coordinate their operations, middleware helps users see the system as a single, cohesive computing environment (figure 1).Therefore, middleware serves as the link between distributed applications running on different hardware platforms, operating systems, network technologies, and programming languages in disparate physical locations [20].

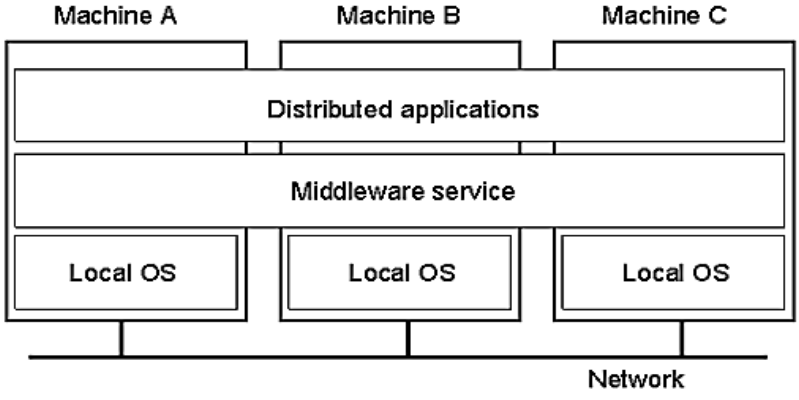

Figure 1. Distributed System [21]

Either the physical components or the user or computation point of view can be used to design the distributed system. The physical perspective is the first, and the logical view is the second. A distributed system is physically made up of a collection of nodes, or computers, connected by a network for communication. The network's nodes do not exchange memory and are only weakly connected. Messages are sent between the system's nodes via the communication network; messages are sent from one node to another using communication protocols. An application's perspective on the system is known as the logical model. It has a number of running processes and channels for communication between them. It is assumed that the core network is completely linked; processes exchange messages with one another. If a system consistently carries out the intended function when properly executed, it is said to be synchronous in a predetermined period; if not, asynchronous. remous, A synchronous system's failure is shown by its inability to respond; consequently, timeout-based methods are employed to identify failures [22].

Networks that are completely or partially linked can be used to build a distributed system. A network that has all of its nodes connected

http://www.doi.org/10.62341/mmle0834

to one another is said to be completely connected. The issue with such a system is that as new nodes are added, the number of nodes linked to the node increases. These results in a significant increase in the complexity and quantity of file descriptors required for each node to implement the connections. Therefore, the ability of each node to open file descriptors and manage the additional connections limits the scalability (capability of a system to continue to operate properly as the system's size or volume changes). Since a message delivered from one computer to another only passes via one connection, the communication cost—the message delay of delivering a message from the source to the destination is minimal. Systems that are fully linked are dependable because, in the event that a few computers or links malfunction, the remaining computers are still able to speak to one another [23].

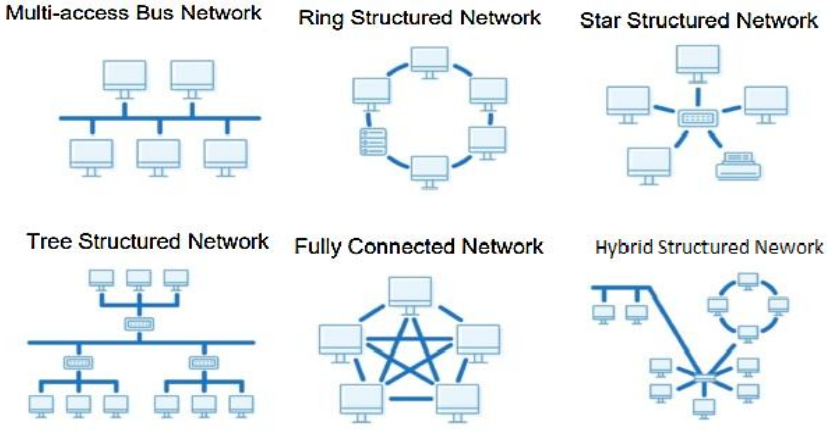

Figure 2. Network Topology

Some computer pairs in a partly linked network have direct connectivity, but not all of them do. Several forms of partly linked networks include tree-structured networks, ring-structured networks, multi-access bus networks, and star-structured networks (figure 2). The network architecture of some conventional distributed systems, like the client/server paradigm, is a star. The issue with such a system is that it will collapse as a whole if the central node fails. A shared communications connection, known as a bus, connects a group of customers in a multi-access bus network. All of the system's nodes cannot connect to one another if the bus link breaks, making it the bottleneck. Performance deteriorates as more machines are added or when there is a lot of traffic, which is another drawback [24].

http://www.doi.org/10.62341/mmle0834

Every node in a ring network is connected to precisely two other nodes, creating a single, continuous signal channel across every node. There is a longer message transmission delay as more nodes are added since the system's diameter increases with the number of nodes. Every node connected to the ring might become isolated in the event of a node failure or cable break. The nodes in a tree-structured network, also known as a hierarchical network, are connected in a tree fashion. Every node inside the network has a set number of nodes connected to it at the hierarchy's next lower level. Since a new node may be added as a child node of the interior or leaf nodes, the tree-structured network has more scalability than a fully linked network; however, only messages sent between a parent node and its child node flow via these systems. Other communications sent between two nodes must pass via one or more intermediary nodes before reaching the destination node [25].

3.2.1.1 Characteristics of a Distributed System

To provide consumers with optimal performance, a distributed system has to have the following features:

Fault-Tolerant: Distributed systems are made up of several software and hardware components that will eventually fail. Service unavailability may result from such component failures; as a result, the systems must be capable of recovering from component failures without taking incorrect action. Fault tolerance aims to prevent system breakdowns even when they do occur in order to maintain uninterrupted operation If a system can conceal the existence of errors, it is considered fault tolerant. Any fault-tolerant system's goal is to make it more reliable or accessible. The likelihood that the system will last until that point is the definition of its dependability [26].

Even in the event that a component fails, a dependable system keeps data safe. The percentage of time that a system is accessible for usage is known as availability. Redundancy is typically used to achieve fault tolerance. The components of the system that are not required for proper operation are referred to as redundancies. There are three categories: time, software, and hardware. Redundancy is accomplished by including other hardware components in the system that, in the event that a component fails, takes over its function in the event that a component fails. Software redundancy consists of additional code and instructions for handling the additional hardware components and utilizing them appropriately

http://www.doi.org/10.62341/mmle0834

for continuous service. The same command is carried out again in temporal redundancy; this is employed to address transient system errors [27].

Scalable: A distributed system may continue to function properly even when a component is expanded in size. Three factors make up scale: the quantity of users and other system elements, the separation between the farthest nodes in the system, as well as the quantity of entities that have administrative authority over certain system components Distributed systems are impacted by the three components of scale in a variety of ways names, authorization, communication, the usage of distant resources, authentication to confirm an individual's identity, and the ways in which users view the system are some of the components that are impacted replication, distribution, and caching are the three methods used to control scale Multiple copies of resources are produced during replication Its application to file, authentication, and name services lessens the strain on individual servers while enhancing the overall dependability and accessibility of the services the two primary replication concerns are the positioning of the copies and the methods used to maintain their consistency the goal of resource replication determines where copies should be placed in a distributed system Replicas are dispersed around the system if a service is being duplicated to minimize network latency when the service is accessed [28].

Predictable Performance: A range of performance indicators, including throughput (the speed at which a network transmits or receives data), reaction time The amount of time that passes between the conclusion of a request or inquiry on a computer system and the start of a response, data, network bandwidth, system usage, etc., are used to evaluate performance. The capacity to deliver the required responsiveness on time is known as predictable performance [29].

Openness: The quality of "openness" guarantees that a subsystem is always available for communication with other systems. Web services are software programs created to facilitate networked, interoperable machine-to-machine communication. These protocols make it possible to grow and expand distributed systems. A scalable open system is preferable to a fully closed, independent system. The feature of openness is attained by a distributed system that is not influenced by the heterogeneity of the underlying environment, including hardware and software platforms. As a result, any client

http://www.doi.org/10.62341/mmle0834

in the system, whether local or distant, has equal access to all services [30].

Transparency: Rather than being seen as a group of collaborating parts, users and application developers should view distributed systems as a whole, the locations of the concurrent operations, and the computer systems used in the operations. Users are not aware of failures, system recovery, resource discovery from various sites, data replication, etc. Transparency allows the system to look and function as a typical centralized system while concealing from its users its dispersed nature. The transparency may be exploited in numerous ways in a distributed system (Figure 3) [31].

Multiple processes can work together utilizing shared information objects without interfering with one another thanks to concurrency transparency (Automatic Teller Machine network, for example). Users won't be aware that there are more system users (even if they consume the same resources); without the users' knowledge, the system can create extra copies of files and other resources for efficiency and/or reliability reasons thanks to replication transparency [32].

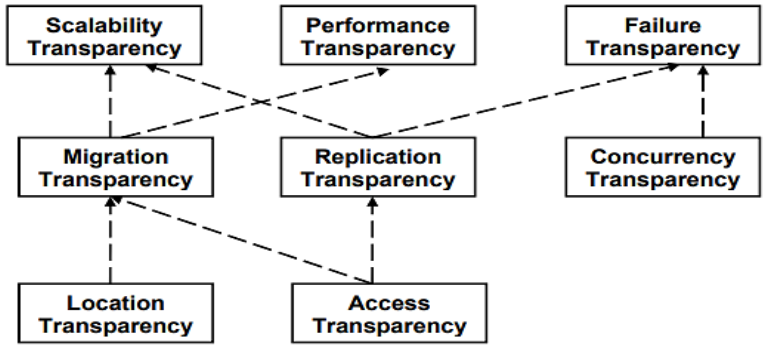

Figure 3. Transparency in Distributed Systems

3.2.1.2 Types of Distributed Computing Systems

Distributed computing systems come in a variety of forms, each intended for a distinct set of applications. These are a few of the most prevalent kinds:

- **Client-server design**: The simplest kind of distributed computing system is this one where several clients are served by a central server over a network.

http://www.doi.org/10.62341/mmle0834

- **Peer-to-peer system design**: This kind of system eliminates the need for a central server by allowing several computers to be linked to one another and share resources and data [33].
- **Grid computing**: This entails several computers cooperating to carry out a challenging activity or solve a big problem. Grid computing is frequently employed in modeling and scientific research.
- **Cloud computing**: This is a distributed computing system that eliminates the need for local hardware or infrastructure by enabling users to access computer resources and services via the internet [34].
- **Computing on the edge**: This entails processing data on network edge devices, such as Internet of Things devices, rather than transferring it to a central server or cloud.
- **Fog computing**: This is comparable to edge computing, except instead of individual devices processing data individually, a network of devices collaborates to do so.
- **Databases that are dispersed**: This entails sharing data among several network nodes in order to increase performance and provide redundancy. Every one of these distributed computing system types has unique advantages and shortcomings and is appropriate for many use cases and applications [35].

3.2.1.3 An Apparent Distributed Environment

A system that looks to be distributed but is actually operating on a single machine or a small cluster of machines is referred to as an apparent distributed environment. The system architecture in such a setting is made to resemble the features of a distributed system, despite the fact that every component operates on the same physical computer or cluster [36]. While avoiding some of the complexity and expense of a genuine distributed system, an apparent distributed environment can offer some of the advantages of a true distributed system, including fault tolerance, scalability, and performance. This method is frequently employed when the development team lacks experience creating and maintaining distributed systems or when the expense and complexity of a fully distributed system are not warranted. A microservices architecture operating on a single computer or cluster is an illustration of an apparent dispersed environment. The system is made up of several tiny, autonomous services that interact with one another over common protocols like messaging or REST in a microservices architecture. Even if all of the services are operating on the same physical machine or cluster,

http://www.doi.org/10.62341/mmle0834

the system may look dispersed if each service is executed in a separate container or virtual machine. A group of database servers set up to function as a single logical database is another illustration of an apparent dispersed environment. This method involves connecting several database servers and sharing data in a manner that seems to be dispersed, even if a single physical computer or cluster houses all of the data. There are many drawbacks to an apparent distributed environment, even if it can offer certain advantages of a real distributed system; for instance, compared to a fully distributed system, it might not be able to offer the same degree of fault tolerance [37].

3.2.2 Directed network protocols

Data transfer between devices in a network is governed by a set of communication standards and guidelines known as directed protocols. Identifying the path or route that data should take to get to its destination is the special emphasis of these protocols by using routing methods to choose the best path depending on variables like network topology, performance, and efficiency. Directed protocols make sure that data is appropriately directed and transported throughout the network. In both local and wide area networks, they are essential for preserving safe, accurate, and dependable data transfer. Internet Protocol (IP) and routing protocols such as RIP, OSPF, and BGP are examples of directed protocols. A key element of contemporary networking is directed protocols, which are developed expressly to regulate the routing and transmission of data between devices connected to a network. Directed protocols are more concerned with making sure that data follows a predefined path based on certain rules and criteria than broad or connectionless protocols. In complicated networks, especially ones that span several devices, segments, or even physical locations, these protocols are crucial. Routing is the fundamental principle of directed protocols. Data packets are guided across a network by a process called routing, which makes sure that each packet gets to its destination accurately and effectively. Directed protocols specify the rules for data transmission, its path, and how it should be addressed [38].

3.2.2.1 Types of Directed Protocols

3.2.2.1.1 Routing Information Protocol (RIP)

http://www.doi.org/10.62341/mmle0834

A kind of distance is used as the hop count metric in RIP, a standardized vector distance routing protocol. It's a vector of distance; RIP avoids routing loops by restricting the amount of hop counts permitted in pathways between sources and destinations. Generally speaking, the most hops permitted for RIP is fifteen; however, the scale of supporting networks is sacrificed in order to minimize routing loops. Since there is a limit of 15 hop counts permitted for RIP, a route will be deemed inaccessible if the number exceeds 15. RIP only sent complete updates once every 30 seconds. When it was initially built, due to the modest size of the routing tables in the early deployments, traffic was not significant. Even if the routers had been launched at different times, significant traffic bursts during the 30-second period are more likely to occur in bigger networks. It is widely accepted that due to this random initialization, in theory, routing changes would be dispersed throughout time, but in reality, this is not the case There are four basic timings in RIP. The router's update timer, which has a default of 30 seconds, determines how frequently it will send out routing table updates when no new information is received on a route; the invalid timer (by default 180 seconds) determines how long the route will stay in a routing database before being flagged as invalid. If an update is received for that specific route before the timer ends, the invalid timer will be reset [39].

The hold-down timer (by default, 180 seconds) shows how long RIP will stop a route from receiving updates while it is in a hold-down condition. RIP will not get any new route updates while in a hold-down state until the hold-down timeout is triggered. A route will be put on hold-down for the following justifications: It's beyond the timer; the route enters a 16 metric (or unreachable); according to an update from another router, the route enters a higher metric than it now uses after receiving an update from another router. Flush Timer: When no new information is received regarding a route, the flush timer indicates how long it may remain in a routing database before being flushed out (by default, it is 240 seconds). The route will be flushed out every 60 seconds after it has been deemed invalid since the flush timers work in tandem with the invalid timer when the routers on the RIP network are not all in sync with the RIP timer. There is system instability. The value of this timer must be greater than the value of the invalid timer [40].

http://www.doi.org/10.62341/mmle0834

3.2.2.1.2 Open Shortest Path First (OSPF)

RFC 2328, an inside gateway protocol used to disseminate routing information within an autonomous system (AS), OSPF is the most popular routing protocol in big business networks out of the three selected samples. OSPF is based on link-state technology and computes the shortest path using the SPF method.

SPF calculation

Prior to doing the computation, each router in the network must be aware of every other router in the same network as well as the connections between them. Finding the shortest route between each router is the next stage Link-states are exchanged for each router which the link-state database would hold Upon receiving a link-state update, a router saves the information in its database and then distributes the new data to all other routers basic illustration of the SPF algorithm's operation may be seen below Five routers make up a basic network (figure 4), and each router is aware of every other router and link. The path details are saved in the link database once every path has been determined [A, B, 3], [A, C, 6], [B, A, 3], [B, D, 3], [B, E, 5], is the link database for the model mentioned above [E, B, 5], [C, A, 6], [C, D, 9], [D, C, 9], [D, B, 3], [D, E, 3], [E, B, 5], and. The terms "originating router," "connected router," and "cost of link between two routers" are all used the router finds the Shortest Path Tree to every destination when each router's database is complete [41].

The Shortest Path Tree

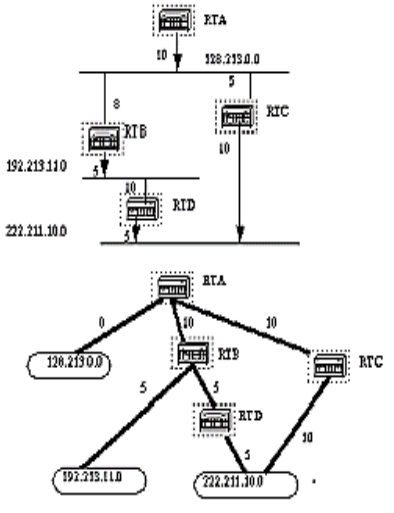

Figure 4. Shortest path tree [42]

http://www.doi.org/10.62341/mmle0834

When the Shortest Path Tree is completed, the router will work on the routing table [43]. An autonomous system can be separated into several portions using the OSPF protocol. An area can be occupied by a segment and a neighboring router. Each segment contains its own database, path tree, and information because they all use the same technique to get the shortest path. They are not visible outside of this area. The database's size may be significantly decreased by doing this. An autonomous system has a backbone known as Area 0, which serves as its pivot. Area 0 used an ABR (Area Border Router) to exchange link states with other sections. Other categories include stub areas, not-so-stubby areas, totally stubby areas, and totally NSSA parts that serve certain purposes [44].

3.2.2.1.2 Enhanced Interior Gateway Routing Protocol (EIGRP)

A hybrid routing protocol called Enhanced Interior Gateway Routing Protocol (EIGRP) offers notable enhancements over IGRP since Internet Protocol is made to handle IPv4 addresses, which IGRP was unable to support. EIGRP took the position of IGRP in 1993 as a hybrid routing system, which was based on the distance-vector protocol but had additional elements of the link-state protocol. It combines the benefits of both Link-State and Distance-Vector routing methods to ensure faster convergence. EIGRP saves all routes instead of just the optimal one; EIGRP only exchanges information that its neighbors would not have, and it maintains nearby routing tables. Unlike outdated distance-vector protocols like RIP, EIGRP is frequently employed in big networks and only updates when the topology changes; the chosen route's optimization is assessed using a metric. Its bandwidth, latency, dependability, load, and MTU form the basis of the EIGRP metric. The EIGRP metric's default expression is $Metric = BandWidth + Delay*$ 256. EIGRP is operated by four fundamental components, which are Discovery/Recovery of Neighbors, Transport Protocol Reliability, Dual Finite State Machine, Module Dependent on Protocol [43].

Having a procedure that allows routers to dynamically learn about other routers on directly linked networks is crucial since EIGRP updates are triggered whenever there is a change. Once a neighboring router is unavailable or malfunctioning, a router should detect it. the neighbor sending brief Hello packets on a regular basis at a little cost allows for discovery and recovery It is possible to ascertain whether this neighbor is still alive once the hello packets

http://www.doi.org/10.62341/mmle0834

are received. When routers are operational, the neighboring router will begin sharing information. Liable transport protocols ensure the assured and ordered delivery of packets for transmission, which is necessary for the default EIGRP algorithm. DUAL A unicast data receiver is sent by EIGRP to indicate that the greeting packet does not need the notification packet's confirmation. A sequence number is assigned during packet transmission, and the router increments this number, delivering a fresh package. Fast transit is ensured by the reliable transit protocol. Even in the case of pending unacknowledged packets, the low convergence time is therefore guaranteed. EIGRP uses the default convergence technique, DUAL, or the Diffusing Update technique, to stop routing loops from recalculating routes. DUAL keeps track of every route and determines the most cost-effective and efficient one, which is then incorporated into the routing table. In the event that the best path is abandoned, there are alternative routes that can be used. Network layer IP packets are encapsulated using protocol-dependent modules (PDM). It uses resources like the routing database to assess whether an extra route is required. PDMs ensure that different routed protocols are supported by EIGRP [44].

3.2.2.1.3 BGP

Traffic between autonomous systems is routed using the Border Gateway Protocol (BGP). A network or collection of networks having shared routing rules and management is called an autonomous system. BGP is the protocol that ISPs use to share Internet routing information. An Interior Gateway Protocol (IGP), like RIP or OSPF, is often used by customer networks, such as those of businesses and institutions, to share routing information among themselves. ISPs use BGP to exchange routes with customers. After customers connect to them, the protocol is known as external BGP (eBGP) when it is utilized between autonomous systems. Interior BGP (iBGP) is the protocol used when a service provider uses BGP to exchange routes within an autonomous system. Since BGP is the routing protocol used on the Internet, it is clear that it is a very reliable and scalable system. BGP employs a large number of route parameters, also known as attributes, to set routing policies and preserve a stable routing environment in order to accomplish scalability at this level. When the neighbors initially establish a TCP connection, BGP neighbors communicate all of their routing information. The BGP routers only communicate the modified routes to their neighbors when they detect changes in the routing

http://www.doi.org/10.62341/mmle0834

table; only the best route to a target network is advertised via BGP routing updates, which are not sent by BGP routers on a regular basis [45].

3.2.2.2 Internet Protocol (IP)

The collection of protocols that the community has at its disposal to enable access by a large number of hosts in a sophisticated and widely dispersed manner is the foundation of Internet accessibility and facilitation. The success of the Internet is largely dependent on the protocols. They are the "software and system agreements" that enable communication between heterogeneous software and hardware over similarly disparate networks. While some innovation permits graphics and limited multimedia, such as audio and video, the present protocols are mostly focused on data transfers. the creation of novel and creative protocols to enable growth for both low-end users and to enhance the information's capabilities will be the problem of the future moved The Transport Control Protocol/Internet Protocol, or TCP/IP protocol suite, is the main underlying protocol stack that enables the Internet to operate through agreements at several network levels to handle, process, manage, and govern the underlying data packets. This protocol enables the quick and simple transfer of data between users. The development of the Internet will be centered on protocols like TCP/IP; later, we'll concentrate on these protocols as they relate to novel access techniques and multimedia. The best way to comprehend the evolution of the protocol is to examine TCP/IP in greater detail. Datagrams can be sent from source hosts to destination hosts via IP (the Internet Protocol), perhaps via one or more routers and networks along the way [41].

A datagram is a packet of bits with a header and a payload that has a limited length in an Internet. The IP headers are processed by both hosts and routers. The hosts need to make sure they are analyzed by the routers in order to make routing decisions, and they must be adjusted as the IP packets travel from their source to their destination. TCP is a protocol designed ... to provide its clients at a higher layer of protocol a reliable, sequenced, flow-controlled end-to-end octet stream. The TCP/IP protocol's twenty-year growth provides the finest insight into how new protocols are developed. Many of the TCP mechanisms' justifications may be comprehended by the following observations: IP merely offers the greatest possible datagram transmission service, whereas TCP functions above it.

http://www.doi.org/10.62341/mmle0834

Sequencing follows end-to-end recovery; for flow control to work, both ends must uniquely agree [46].

3.2.2.3 IP Telephony Strategic Alternatives

offering a worldwide IP network with a quality of service (QoS) guarantee for a cost commensurate with the service's quantity, and perhaps actual usage, the provider can take over as the network backbone at the core thanks to the IP service network strategy at the IP level, and that, contrary to what is done now, everyone who wants to connect to them must do so at the IP level rather than the raw bandwidth level. Such an IP service offering is aimed at the local IP service provider rather than the carrier level in the present Internet provider environment with service expenses above the EBIDTA line rather than below it as a capital asset; it is nearly a "reseller" strategy [47].

3.2.2.4 MANET

Mobile Ad hoc Network is referred to as MANET. It is a strong wireless network without any infrastructure. Mobile nodes alone or in combination with fixed and mobile nodes can constitute a MANET. Nodes randomly connect to one another to build arbitrary topologies; they serve as both hosts and routers. Mobile routers' self-configuring capabilities make this technology appropriate for conferences, emergency search and rescue missions where a network connection is desperately needed, and disaster-affected areas without communication infrastructure. The Internet Engineering Task Force (IETF) established the MANET working group in order to provide uniform IP routing protocols for wireless networks due to their necessity for mobility topologies that are both static and dynamic. Despite years of development, there is still no fully developed Internet standard for MANET protocols. Only since 2003 have experimental Requests for Comments (RFCs) been identified. At this point, although it appears that there are still unresolved issues with the protocols' deployment or implementation, the suggested algorithms are recognized as experimental technologies, and there is a good probability that they will become the norm. Since then, this field has seen a surge in study, with notable works on Temporally Ordered Routing Algorithm (TORA), Dynamic Source Routing (DSR), Ad hoc On-demand Distance Vector (AODV), and Optimized Link State Routing (OLSR) [48].

http://www.doi.org/10.62341/mmle0834

Routing Protocols in MANETs

A standard for managing node choices during packet routing over a MANET between devices is known as an ad hoc routing protocol; a node in the network or one attempting to join is unaware of the network topology. The topology is found using declaring its existence and keeping an ear out for broadcasts from other network nodes, or neighbors. The routing protocol used in a network determines how the route discovery process is carried out. For wireless ad hoc networks, a number of routing protocols have been developed. There are two types of routing protocols: proactive and reactive. Ad hoc routing methods can combine reactive and proactive features; in some cases, they are known as hybrids [49].

3.2.2.5 Proactive Routing Protocols

Routing information is created and maintained for every node by proactive routing protocols. Whether the route is required or not has no bearing on this. Control messages are sent out on a regular basis to do this routing. Methods that are proactive are not bandwidth-efficient; this is because, even in the absence of data flow, control messages are transmitted. There are benefits and drawbacks to this kind of routing mechanism. The ease with which nodes may obtain routing information and create a session is one of its primary benefits. The drawbacks include the fact that the nodes store an excessive amount of data for route maintenance and that it is sluggish to restructure in the event that a specific connection fails. One instance of a proactive routing protocol is the Optimized Link State Routing Protocol (OLSR) [50]**.**

3.2.2.6 Dynamic Source Routing

DSR is an ad hoc wireless network reactive routing technology. Although it is not table-driven, it also contains on-demand features similar to AODV. The foundation of it is source routing; the path for a packet is specified by the node that wishes to send it. The complete route details of the packet traveling across the network from its origin to the sender specify the destination in the packet. This kind of routing differs from table-driven and link-state routing in how routing choices are chosen. the source node makes the routing decisions in source routing, The addresses of every intermediary node between itself are gathered by the source node as well as the final destination when identifying routes. All of the nodes participating in the route discovery process store the path information gathered by the source node [51].

http://www.doi.org/10.62341/mmle0834

4. Results and discussion

Routed protocols in dispersed networks are studied to show how response speed is affected by different routing strategies under varied network circumstances. How fast data can move over a network depends on a number of factors, including protocol type, network topology changes, network size, algorithm efficiency, and overhead. Each of the three main categories of routing protocols proactive, reactive, and hybrid has a distinct impact on response time in stable network conditions. Proactive protocols like OSPF ensure quicker answers by keeping routing tables current; however, when the network topology is dynamic, this constant exchange of routing information results in slower replies and more overhead. Reactive protocols, such as AODV, on the other hand, only create routes when necessary. Although route finding causes longer initial response times, this reduces overhead and improves efficiency in networks with frequent topology changes. Depending on the status of the network, hybrid protocols provide a balance between proactive and reactive components, resulting in reasonable response time. Another important aspect influencing reaction speed is the scale of a dispersed network. Proactive protocols are effective in smaller networks because they can keep routing tables current without adding a lot of overhead. Slower response times result from the increased resource requirements of maintaining these routing tables as the network expands (table 1).

Table 1: Effect of Routing Protocol Type on Response Speed

Routing Protocol Type	Response Speed	Advantages	Disadvantages
Proactive (e.g., OSPF)	Faster in stable networks	Maintains up-to-date routes, reducing delays in stable conditions.	High overhead for maintaining routes, slower in dynamic conditions.
Reactive (e.g., AODV)	Slower initial response but faster after route discovery	Lower overhead, suitable for dynamic environments.	Initial delay during route discovery, higher latency in unstable conditions.
Hybrid (e.g., ZRP)	Balanced response speed depending on network state	Combines benefits of proactive and reactive protocols, adaptable to changes.	May introduce some complexity and additional processing time in large networks.

http://www.doi.org/10.62341/mmle0834

Reactive and hybrid protocols are frequently chosen in bigger networks due to their superior scalability. Reactive protocols are more effective in large-scale or highly dynamic situations because they require less maintenance, even though they are slower at first in finding pathways with a solid balance between scalability and reaction time. Hybrid protocols, which include proactive and reactive techniques, perform effectively in medium-to-large networks.

Proactive protocols like OSPF or IS-IS are advised for networks that are stable and comparatively static since they can keep routes current and provide prompt replies to data requests. These protocols are perfect for networks with few topological changes since they minimize the latency in route finding. However, reactive protocols like AODV or DSR should be taken into consideration in extremely dynamic situations where nodes join or depart regularly. By only finding routes, when necessary, these protocols minimize overhead and improve the network's response time under dynamic circumstances.
Response times may be delayed by high overhead, especially in proactive measures. Periodically optimizing the frequency of route updates in proactive protocols is advised to lessen this, particularly in big networks. Response times can be shortened by lowering the frequency of route updates, which will also reduce the volume of control traffic. To further cut down on overhead, hybrid protocols can be configured to function in a reactive mode when there is a lot of mobility or when there are significant topology changes. Hybrid algorithms, which balance proactive and reactive routing mechanisms, are crucial for large networks with unpredictable topology changes, ensuring optimal performance and avoiding excessive computational load.

5. Conclusion

By affecting the effectiveness of data transmission and processing across several nodes, routed protocols significantly impact the reaction speed of dispersed networks. Network stability, latency, and bandwidth use are all strongly impacted by the selection and architecture of these protocols. Reliable communication is ensured by protocols like TCP/IP, which rely on strong error-checking and retransmission procedures. However, in high-latency contexts, these protocols may cause delays. Lightweight protocols, on the other

http://www.doi.org/10.62341/mmle0834

hand, could increase speed but run the risk of losing data or becoming less reliable. Protocol design may maximize reaction speed based on the dynamic conditions of the network, as demonstrated by the balance between proactive, reactive, and hybrid routing algorithms in networks such as MANETs. For example, proactive protocols guarantee quick data transmission in networks with consistent connectivity, whereas reactive protocols minimize needless overhead in low-traffic scenarios. The constant development of routed protocols, adapted to the unique requirements of cutting-edge technologies like cloud computing, multimedia streaming, and the Internet of Things, is ultimately necessary to increase the reaction speed of distributed networks. Protocol algorithm innovations, such as improved quality of service (QoS) mechanisms and adaptive routing, are crucial to meeting the increasing demands for distributed systems that are quicker and more dependable.

References

[1] Kamal Shahid, Saleem Naseer Ahmad, and Syed Tahir Hussain Rizvi. (2024) "Optimizing Network Performance: A Comparative Analysis of EIGRP, OSPF, and BGP in IPv6-Based Load-Sharing and Link-Failover Systems," September 2024, 16(9):339

[2] Sudipto Ghosh, Aditya P. Mathur. (1999), "Issues in Testing Distributed Component-Based Systems", Software Engineering Research Centre, West Lafayette, March 1999.

[3] Masruroh, S.U.; Robby, F.; Hakiem, N. (2016) Performance evaluation of routing protocols RIPng, OSPFv3, and EIGRP in an IPv6 network. In Proceedings of the 2016 International Conference on Informatics and Computing (ICIC), Mataram, Indonesia, 28–29 October 2016; pp. 111–116.

[4] Ali, A.B.; Tabassum, M.; Mathew, K. (2016) A comparative study of IGP and EGP routing protocols, performance evaluation along with load balancing and redundancy across different AS. In Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong, China, 16–18 March 2016; Volume 2, pp. 487–967.

[5] A. Bright, M. Adamu, A. Franklin, and M. Asante, (2016) "Performance Analysis of Enhanced Interior Gateway Routing Protocol (EIGRP) Over Open Shortest Path First (OSPF)

http://www.doi.org/10.62341/mmle0834

Protocol with Opnet," Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 5, pp. 77–82, 2016, doi: 10.14569/ijacsa.2016.070512.

[6] P. Rakheja, P. Kaur, A. Gupta and A. Sharma, (2012) "Performance Analysis of RIP, OSPF, IGRP and EIGRP Routing Protocols in a Network", International Journal of Computer Applications, vol. 48, no.18, (2012), pp. 975-888.

[7] Thorenoor, S.G. (2010) Dynamic Routing Protocol Implementation Decision between EIGRP, OSPF, and RIP Based on Technical Background Using OPNET Modeler" (Wipro, Bangalore, India). Source: Proceedings of the 2010 Second International Conference on Computer and Network Technology (ICCNT 2010), p191-5, 2010. Renata Teixeira, Jennifer Rexford, "Managing Routing Disruptions in Internet Service Provider Networks," IEEE Communications Magazine, March 2006.

[8] Dong Xu, (2011) "Analysis of OSPF, EIGRP, and RIP protocols for real-time applications." http://www.sfu.ca/~donx/" SFU proceedings Spring 2011

[9] Nohl, A.R., Molnar, G. (2002) "The convergence of the OSPF routing protocol" (Ericsson Res., Ericsson Hungary Ltd., Budapest, Hungary); Source: Periodica Polytechnica Electrical Engineering, v 47, n 1-2, p 89-100, 2002.

[10] Wijaya, Chandra. (2011) "Performance Analysis of Dynamic Routing Protocol EIGRP and OSPF in IPv4 and IPv6 Network." In Informatics and Computational Intelligence (ICI), 2011 First International Conference on, pp. 355-360. IEEE, 2011.

[11] Massively Distributed Systems: Design Issues and Challenges, Dan Nessett, Technology Development Center, 3Com Corporation. Proceedings of the Embedded Systems Workshop, Cambridge, Massachusetts, USA, March 29–31, 1999

[12] Scheduling Problems (2005) for a Class of Parallel Distributed Systems, Hiroshi Tamura, Futoshi Tasaki, Masakazu Sengoku, and Shoji Shinoda Niigata Institute of Technology, Japan, Faculty of Engineering, Niigata University, Japan, IEEE 2005.

[13] Distributed Computing: Principles, Algorithms, and Systems, Chapter 10, Ajay Kshemkalyani and Mukesh Singhal.

[14] Distributed software Page: 479-501. Engineering, Ch-10, Ian Sommerville,

http://www.doi.org/10.62341/mmle0834

[15] Monitoring Distributed Systems, Masoud Mansouri-Samani and Morris Sloman, Imperial College.

[16] Jing Deng, S.C.-H. Huang, Y.S. Han, J.H. Deng, (2010) Fault-tolerant and reliable computation in cloud computing, in: 2010 IEEE GLOBECOM Workshops (GC Wkshps), 6–10 Dec. 2010, pp. 1601–1605.

[17] A. Acharya, M. Ranganathan, and J. Saltz,(1997) "Dynamic linking for mobile programs, In Mobile Object Systems: Towards the Programmable Internet", pp. 245-262, Springer-Verlag.

[18] Zapf M Herrmann K. GeihsK (1999), Decentralized SNMP Management with Mobile Agents, Proceedings of 6th Int. Symposium on Integrated Nwk Mer, Boston, MA USA, 24-28 May, 623-635

[19] Bernstein P (1996) Middleware: a model for distributed system services Commun ACM 39(2):87-98

[20] Blair G, Stefani J-B (1998) Open distributed processing and multimedia. Addison-Wesley, Reading

[21] TANENBAUM, A.S., and VAN STEEN, M. (2007) Distributed Systems: Principles and Paradigms, Second Edition, Upper Saddle River, NJ: Prentice Hall, 2007. pp 22

[22] Ben-Ari M (2006) Principles of concurrent and distributed programming, 2nd end. Prentice Hall, Englewood Cliffs

[23] Baset S. Schulzrinne H (2006) An analysis of the skype peer-to-peer internet telephony protocol. In: 25th INFOCOM Conference, IEEE, IEEE Computer Society Press, Los Alamitos, CA, pp 1-11

[24] Baldauf M, Dustdar S, Rosenberg F (2007) A survey on context-aware systems. Int J Ad Hoc Ubiquitous Compute 2:263-277

[25] Bonnet P, Gehrke J, Seshadri P (2002) Towards sensor database systems. In: Second international conference mobile data management. Springer, Berlin. Lecture notes in computer science, vol 1987, pp 3-14

[26] Adelstein F, Gupta S, Richard G, Schwiebert L. (2005) Fundamentals of mobile and pervasive computing. McGraw-Hill, New York

[27] Akyildiz IF, Kasimoglu IH (2004) Wireless sensor and actor networks: research challenges. Ad Hoc Netw 2:351-367

http://www.doi.org/10.62341/mmle0834

[28] Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) A survey on sensor networks. IEEE Commun Mag 40(8):102-114
[29] Akyildiz IF, Wang X, Wang W (2005) Wireless mesh networks: a survey. Comp Netw 47(4):445-487
[30] Alonso G, Casati F, Kuno H, Machiraju V (2004) Web services: concepts. Springer, Berlin
[31] Amar L, Barak A, Shiloh A (2004) The MOSIX direct file system access method for supporting scalable cluster file systems. Cluster Computers 7(2):141-150
[32] Amza C, Cox A, Dwarkadas S. Keleher P, Lu H, Rajamony R, Yu W, Zwaenepoel W (1996) Trademarks: shared memory computing on networks of workstations. IEEE Computers 29(2):18-28
[33] P. Huang, Y. Cao, and Y. Wang, (2021) "A Distributed Task Assignment Scheme for Mobile Edge Computing," IEEE Transactions on Mobile Computing, vol. 20, no. 1, pp. 72-84, Jan.
[34] D. Farias, M. H. P. Chaves, and A. Loureiro, (2021) "Scalable Distributed Deep Learning: Challenges and Opportunities," IEEE Internet Computing, vol. 25, no. 1, pp. 68-75, Jan./Feb.
[35] D. Chen, J. Li, and J. Huang, (2021) "Elastic Distributed Inference for Deep Learning: A Comprehensive Survey," ACM Transactions on Parallel Computing, vol. 7, no. 1, Article.
[36] Bader, David; Pennington, Robert (2001). "Cluster Computing: pplications". Georgia Tech College of Computing. Archived from the original on 2007-12-21. Retrieved 2017-02-28.
[37] Z. Zhang, Y. Chen, and M. Chen, (2021)"Distributed Computing for Deep Learning: A Review, "ACM Transactions on Intelligent Systems and Technology, vol. 12, no. 1, Article 12.
[38] H. Zheng, Z. Liu, and Y. Chen, (2021) "Distributed Optimization with Randomized Algorithms for Large-Scale Multi-Agent Systems," IEEE Transactions on Cybernetics, vol. 51, no. 4, pp.
[39] Behrouz A. Forouzan, (2009) “TCP/IP Protocol Suite”, McGraw-Hill Education Press. P. 269. ISBN 0-073-376043.

http://www.doi.org/10.62341/mmle0834

[40] Pankaj Rakheja, Prabhjot Kaur, Anjali Gupta, Aditi Sharma, (2012) "Performance Analysis of RIP, OSPF, IGRP and EIGRP Routing Protocols in a Network.

[41] Scott M. Ballew, (1997) "Managing IP Networks with Cisco Routers", O"REILLY Press. Chapter 5. ISBN: 1-56592-320-0.

[42] Adarsh Kumar, (2021) "How Open Shortest Path First (OSPF) uses Dijkstra's Algorithm for finding Shortest Routing?", cloud technical solutions engineer, Jul 10, 2021

[43] B. Wu, (2013) "Simulation Based Performance Analysis on RIPv2, EIGRP and OSPF Using OPNET.

[44] Hubert Pun, (2001) "Convergence Behavior of RIP and OSPF Network Protocols".

[45] Behrouz A. Forouzan, (2009) "TCP/IP Protocol Suite", McGraw-Hill Education Press. P. 269. ISBN 0-073-376043.

[46] Jeff Doyle, (1997) "Routing TCP/IP (Volume I)", Cisco Systems Press. Chapter 5-9.

[47] Hubert Pun, (2001) "Convergence Behavior of RIP and OSPF Network Protocols".

[48] Vahid Nazari Talooki and KoorushZiarati, (2006) "Performance Comparison of Routing Protocols for Mobile Ad Hoc Networks" Asia-Pacific Conference on Communications, APCC, pp. 1 – 5

[49] D. Kiwior and L. Lam, (2007) "Routing Protocol Performance over Intermittent Links" Military Communications Conference, MILCOM, IEEE, pp. 1 – 8

[50] S. Demers and L. Kant, (2006) "MANETs: Performance Analysis and Management", Military Communications Conference, MILCOM, pp.1 – 7

[51] Jin Mook Kim, In Sung Han, Jin Baek Kwon, Hwang Bin Ryou, (2008) "A Novel Approach to Search a Node in MANET", Information Science and Security, ICISS, pp. 44 – 48