JAUES

Journal of Al-Azhar University Engineering Sector

Vol.16, No. 61, October, 2021,1156 -1170

DESIGN AND INSTALL SECURE AND SCALABLE PRIVATE CLOUD PHONE

Tamer ELNawawy [1], Khalil Mohammed [1], Hany Harb [1]

[1] Systems & Computers Engineering Department, Faculty of Engineering, Al-Azhar University, Cairo, Egypt.

*Corresponding Author E-mail:Tamer.Nawawy@gmail.com

Received : 18 August 2021 Accepted: 14 Sept 2021

ABSTRACT:

In the last declining years, cloud computing is instituted as internet-based computation to offload massive computing to remote servers and offer on request network access to a shared pool of customizable servers, memory storage, networks, and applications by interlink physical and virtual appliances. Cloud computing has arisen as a novel trend in VoIP (Voice over Internet Protocol technology). It enables and helps in reaching high availability to perform scalable and secure VoIP systems, share knowledge, and storage wide knowledge effectively. The usage of the cloud appends some other perspective to these subjects with the load balancing of virtual machines because of the uninterrupted usage of the minimal IP routing upon the same destination. Furthermore, the utilization of customary equipment, protocols, and security structures that cannot sufficiently guard VoIP frameworks against recently made malicious assaults like DDoS and others. The goal of this research is to provide a scalable and secure Private Cloud phone that relies upon dynamic load balancing to allow calls to the recipient by estimating the usage of clustered virtual machines, selecting requests to the least loaded VM with the fastest processing time and least load to guarantee null delay, packet loss, and best performance. Besides, the proposed framework provided extra techniques to detect DDoS attacks and eliminate their risks to this model. The NS2 and Cloud-Sim check systems were utilized to join a cluster of SIP Proxy servers in our model. The applied technique draft a checked improvement in a few major DDoS parameters; that is, in regards to the detection rate, and usually VoIP performance.

KEYWORDS: VoIP; Cloud phone, SIP servers, DDOS, and load balancing

تصميم نظام مكالمات عبر الانترنت يعمل من خلال منظومة الحوسبة السحابية

واتخاذ الاجراءات اللازمة لتأمينه وتوزيع الأحمال بطريقة قياسية

تامر النواوي[1]، خليل محمد[1]، وهاني حرب[1]

[1]قسم هندسة النظم والحاسبات, كلية الهندسة, جامعة الازهر, القاهرة, مصر

* البريد الاليكتروني للباحث الرئيسي: Tamer.nawawy@gmail.com

الملخص

في السنوات الأخيرة ، ظهرت منظومة الحوسبة السحابية على شبكة المعلومات الدولية (الإنترنت) لتوفير عدد من الخدمات الحاسوبية المتكاملة دون التقيد بالموارد المحلية بهدف التيسير على المستخدم ، وتحسين آداء الخوادم الرئيسية واستغلال موارد الشبكة وتخزين الذاكرة ، وتقديم افضل التطبيقات عن طريق ربط الأجهزة المادية والافتراضية. نشأت فكرة انشاء المكالمات عبر الحوسبة السحابية كإتجاه جديد فى بروتوكول الإنترنت لتطوير تقنيات المكالمات والاتصالات وتحسين خدماتها بدلا من استخدام الطرق التقليدية فى تلك الأنظمة. فهي تمكن وتساعد في الوصول لأداء أفضل لأنظمة تقنيات الصوت أكثر تطوير وآمنة وقياسية، وأكثر مشاركة للمستخدين ، مع توفير سعات تخزين فعالة. يرتبط استخدام منظومة الحوسبة السحابية ببعض المحددات الأخرى من بينها استخدام أجهزة افتراضية كهدف للمستخدمين لتحقيق تلك المكالمات من خلال IP مما أظهر ضرورة تأمين تلك الأجهزة . علاوة على ذلك ، استخدام المعدات والبروتوكولات والهياكل الأمنية المعتادة التي لا يمكنها حماية أطر منظومة المكالمات عبر الانترنت بشكل كاف ضد الهجمات الخبيثة التي تم إجراؤها مؤخرًا مثل DDoS وغيرها. الهدف من هذه البحث هو توفير منظومة هواتف عبر الحوسبة السحابية الخاصة بوزارة الداخلية قابل للتطوير وآمنة من خلال موازنة الحمل الديناميكي للسماح بتلك المكالمات للوصول للمستلم من خلال تقدين نظام متطور لتوصيل المكالمات للخادم المناسب ، الأقل تحميلًا مع أسرع وقت للمعالجة وأقل حمل لضمان عدم تأخير المكالمة ، وتقليل فقدان لحزم الصوت ، مما يسهل فى الوصول لمعدلات اداء عالية. إلى جانب ذلك ، تقديم تقنيات إضافية لاكتشاف هجمات DDoS والقضاء على مخاطرها على هذا النموذج. تم استخدام أنظمة فحص NS2 و-Cloud Simللانضمام إلى مجموعة من خوادم SIP Proxy في نموذجنا. تقوم التقنية المطبقة بتقديم نظام جديد قياسى ومؤمن تغلب على صعوبات وعيوب فنية بطرق سابقة تم استخدامها فى نفس المجال.

الكلمات المفتاحية : مكالمات عبر بروتوكول الانترنت , الحوسبة السحابية , خوادم الهواتف, رفض الخدمات الموزعة , توزيع الأحمال

INTRODUCTION

Clouds are a big pool of smoothly serviceable and available virtualized resources (such as hardware, establishment platforms, and/or infrastructure). Due to The National Institute of Standards and Technology (NIST), the five vital characteristic features of Cloud are on request self-service, extensive network admission, resource sharing, quick flexibility, and gauged servicing. Eventually, the catalog four potential utilization models for cloud computing: private cloud, community cloud, public cloud, and hybrid cloud [1]. There are three primary kinds of Clouds. The Public Cloud drive on an embankment of virtual servers situated off-site, and the resources are joint by several customers and consulted via the wideband. A Private Cloud is typically functioning in-house and occurs rear the institution firewall [2]. Resources of Private Clouds are not shared out by any other reception. A Hybrid Cloud is a combining of resources outsourced to third parties, with the majority of the data banking and framework outstanding at household, rear the security of a trusted firewall. Infrastructure as a Service (IAAS), Platform as a Service (PAAS), and Software as a Service (SAAS) are the three major layers of Cloud computing systems. Cloud computing faces many challenges such as security, effective load balancing, performance monitoring, consistency, resource scheduling, scalability, quality of service management, virtual machine migration, fault tolerance, server integration, and high availability [3].

Load balancing is a process to accomplish proficient utilization of resources by reassigning an absolute load of individual cluster nodes and to improve task response time, which is a significant issue, and disseminates the workload across numerous nodes to guarantee that no single resource is overloaded or not utilized totally [4]. There are several major objectives of load balancing such as Cost-effectiveness, Scalability and flexibility, Priority, Avoiding bottlenecks, reducing response time, increasing the performance significantly, Keeping the system stable, and Load balancing strategies [5]. There are two fundamental types of Load balancing. The first type is static load balancing which is characterized by the design of the framework. System pre-knowledge is already known. The principal disadvantage of this type is that it does not take the status of the framework while settling on assignment choices, which has a major impact on the overall performance of the system. The second type is dynamic load balancing which measures only the status of the framework during load-balancing choices to allocate the calls to the lighter servers. There are distributed and centralized dynamic load balancing [6].

Voice over IP (VoIP) implies voice is transmitted over a digital network. A cloud phone system is a communication environment where applications stay on the cloud/remote site kept up, oversaw by the telephone service provider, and meet all of the earnest necessities of customers. Since VoIP utilizes packets, more data can be transmitted over the network to help and upgrade communication needs when compared with traditional communication strategies. Cloud phone faces challenges like security, flexibility, and reliability [8]. VoIP signal is transformed into the digital structure by an analog-to-digital converter, voice data is packetized, and encoded preceding transmission of the signal at the sender has appeared in **Fig.1. Fig.2** illustrates the structure of VoIP packet [7].

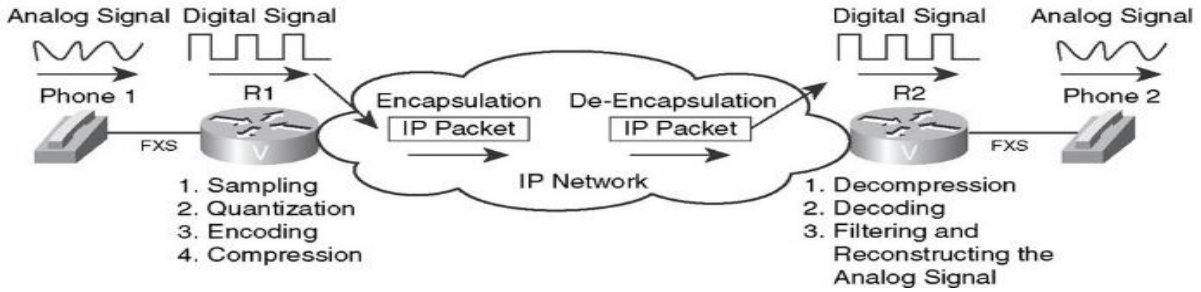


Fig. 1: VoIP Digitizing and Packetizing

Ethernet-link layer header (18 bytes)	IP header (20 bytes)	UDP header (8 bytes)	RTP header (12 bytes)	G.711 codec voice Payload (160 bytes)

Fig.2: VoIP packet format

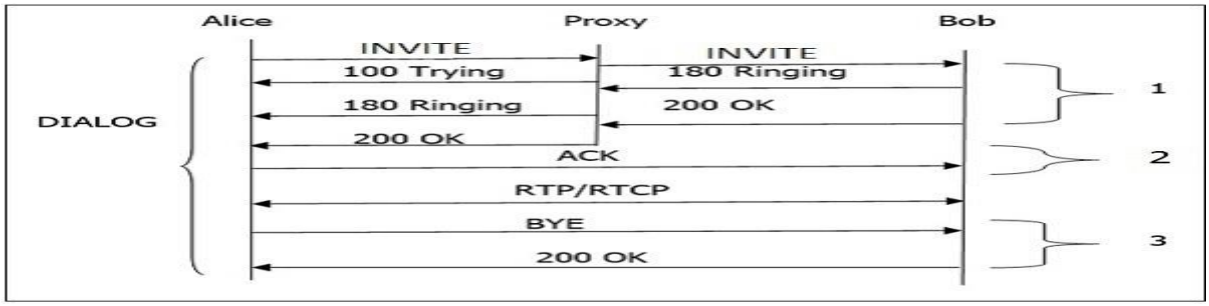


Fig. 3: Basic SIP Operation

SIP operation comprises two SIP telephones utilizing a SIP proxy server to route calls to obscure destinations. Caller telephone (Alice) begins by sending an 'INVITE' request to the callee telephone (Bob) including SDP data for the mentioned session type. Since it doesn't know the first thing about the IP address of the callee phone, it passes on the signal to the proxy server that sends a '100 trying' response back to the guest phone disclosing to it that the proxy server is endeavoring to route the call. The proxy server sends the 'INVITE' request from caller telephone to callee telephone that begins to ring in the wake of connecting with the Registrar server to know the subtleties and the location of the callee, and afterward telling the client of an incoming call. Callee telephone sends a '180 RINGING' response back to the caller telephone through the proxy server and the caller telephone likewise begins to ring, showing that the call is being set up. At the point when the callee lifts the handset and answers the call, the callee telephone sends a'200 OK' response to the caller telephone employing the proxy. The '200 OK' message contains the SDP media description that notices to the guest phone what kind of media meeting that the callee phone can manage for the call. The guest phone sends an 'ACK' message clearly to the callee phone and the bidirectional RTP media stream is set up enabling guest and callee to converse with each other. At the point when the discussion is finished by any of the members, the ending gathering's telephone sends a 'BYE' request to the next part which response with a '200 OK' message that ends the RTP stream, and the call is ended. See Fig.3 [7].

Cloud phones have various benefits such as fewer infrastructure management headaches and expenses, portable Connection, Greater control, Outstanding Reliability, Scalability, Integration with clients, easy Software, Desktop, and Mobile Applications, Increased productivity, Enhanced usefulness, QoS Parameters of VoIP Traffic. Etc. VoIP signals are changed and transferred over into IP packets that may go over many networks access points. Along these lines, the data is introduced to various potential attack centers that can be used to catch hackers. All the security hazards related to IP, for example, PC infections, Denial of Service, and man-in-the-middle attacks are dangerous to VoIP systems. The effects of VoIP attacks can go from inferior quality of service to general setback (in any case called call dropping) for help. There are four main types of VoIP attacks: Attacks against availability, Attacks against confidentiality, Attacks against Integrity, and Attacks against social context [9].

1. LITERATURE REVIEW

1.1. Dynamic Methods for Load Balancing in Cloud Computing

Nikita Haryani A et AL [10]. proposed a distributed dynamic load-balancing algorithm based on measuring the current workload at each node. It checks the counter variable of every server node and virtual machine. The counter variable is the number of requests to access the cloud VoIP application that the particular server node is currently handling. Then, it transfers the load by choosing the minimum value of the counter variable. This strategy has several disadvantages: The discretionary transfer of load causes some servers to be seriously loaded while various servers are gently loaded. The system could not distribute the load correspondingly which doesn't improve the performance or decrease the time delay. The response to request ratio is not considered. The approach did not take into consideration the parameters of response time and the processor utilization, which have, remarkable on the total performance not only the load on each virtual node.

Anita et al [11]. recommended a Centralized dynamic load balancing Technique dependent on the dispatcher. It verifies the incoming VoIP request data according to which a specific server providing that service is allocated. They rank the Web services based on the highest hit rate of the service. When it receives requests from users, it keeps incoming requests in a queue. It analyzes the type of content requested, gets a list of clusters, and searches the content matching ranking cluster. It finds the load status of each server and locates the server with a minimum load. This methodology has several disadvantages. It prompts a particular overhead on the dispatcher. The content-based framework takes additional time than a non-content-based framework to serve the requests. It did take into consideration the parameters of response time and the processor usage, not only the load on each virtual node.

Sunny Nandwani et al [12]. presented a weight-based data center selection algorithm that assigns weights to each data center, depending on the number of VMs. The results gave that the data center with fewer numbers gives better performance. The results show that the cost of this algorithm has a negligible difference with random selection is done in service proximity-based routing. This policy has several disadvantages. Improving the financial cost and power consumption is still to be researched and improved especially with the expected increase of VMs. The proposed policy requires further improvements especially in the case of variable DC's cost or power consumptions.

Venkateshwarlu Velde et al [13]. proposed an optimized weighted round-robin approach. The tasks running in VMs in cloud computing may be of two different types. They are pre-emptive and non-preemptive. Preemptive tasks are released automatically when the given time is over in a round-robin fashion. In non-preemptive tasks, it is difficult to force a process to stop work on the task and release control, which causes a problem for smooth running. They designed a scheduler that has its own logic to assign tasks to VMs appropriately. The load balancer migrates tasks from a heavily loaded VM to a lighter VM. When there are non-preemptive dependent tasks, it handles them properly without the need for migrating to other VM. The approach has some defects: It did not enhance interesting parameters which if they were taken into consideration in the provided algorithm that would improve more and more in the performance of the system resources. Randomly migration of loaded VM to any non-loaded VM causes some servers to heavily load more than the other. The approach did not take into consideration some other parameters like response time and processing utilization.

Mohammad Alhihi et al [14]. utilized the Multi-Path labeling framework Traffic Engineering that uses encapsulated fixed-length labels to specify the optimum number of independent shortest paths. They

proposed a way for finding the set of shortest paths based on the mathematical concepts. This method has cons: First, Lack of Total Control–The service provider needs to configure the general networks. There is a need to work alongside the provider in routing MPLS traffic. MPLS does not permit having absolute control of the network. Besides, Expensive – Since MPLS can cost more than Ethernet.
Faritha Banu et al [15]. designed an algorithm for load balancing to manage the Flow classification. When the stream goes into the MPLS edge node, the MPLS packet classifier recognizes the label field, expecting the name is not assigned. The Flow classification algorithm considers the threshold value for packet loss rate, and stream arrival rate. There are likewise a few downsides to utilizing MPLS. Initially, MPLS is streamlined for highlight point availability as it were. They are not ideal for cloud clients. We cannot straightforwardly get to SaaS or cloud applications with MPLS. Second, MPLS will set aside a long effort to send distinctive data centers on the world. MPLS likewise requires WAN improvement for streamlining its delivery, which will include additional costs on top of the old expense.
Jorge M. Cortés-Mendoza et al [16]. dissected genuine data assembled during one month of MIXvoip association service to oversee realistic VoIP cloud conditions. It was showed that load-aware strategy with expectations strategies outperforms the realized ones giving reasonable quality of service and lower cost. Call allocation strategies are grouped by the type and amount of information used for allocation: (1) knowledge-free (KF), (2) utilization-aware (UA), (3) time-aware (TA), and (4) load-aware (LA). One of the cons of this approach, the authors did not combine these strategies together to improving the VoIP parameters.

1.2.Methods for protecting VoIP systems against malicious attacks

Abualhaj et al [17]. proposed architecture would put the SIP server on Microsoft Azure's cloud with security layer protection addressed by IPS that reject pernicious traffic before passing to the SIP server. This methodology has a few disadvantages. First, it negatively affects the performance of the SIP VoIP server. Furthermore, it forced an extra traffic delay. Third, the bottleneck occurred in the call signaling. Fourth, Other VoIP signaling protocols, for example, Inter-Asterisk eXchange (IAX) and H323 in the cloud have not been compared and the security layer protection. Finally, they should unmistakably depict the design architecture methodologies used in the original copy.
Kolhar et al [18]. completed a VoIP application on a VM on Microsoft Azure Cloud. The firewall was used and Intrusion Detection System (IDS) was presented on SIP Proxy Server, to analyze and get risks against it as parts of the Proxy server. The technique brought about some flaws such as, the deep detection and reduction technology DDOS not mentioned, moreover they used anti-virus software that could not detect all viruses and they did not explain how to program gateways to accept network traffic of certified IDS agents working with SIP clients on the customer site. This methodology may suit few VoIP clients but not an enormous number of clients or calls.
Chauhan et al [19]. recommended installing big data for Machine learning to make a dataset addressing SIP traffic in reality with countless RTP packets in online protection solutions for mechanizing DDOS discovery in SIP-based VoIP networks, recognize anomalies from typical traffic, diminish detection times, stay away from bogus recognitions, address failover and gives custom-fitted load balancing. This original copy contains numerous muddled focuses. The program utilized in the exploration was not formed as clear strides for the directions utilized. The recently referenced strategy is questionable and has not been clarified. The author has not referenced any issues with the procedures for introducing big data for AI. The dataset did not have clear properties. DDOS detection in SIP-based VoIP networks did not have specific parameters. The specific measure of location time was not referenced. The avoidance of false detection and its strategies were not recognized. The load balancing was planned without the depiction of its highlights, types, methods, norms ... and so forth.
Jeyanthi and Iyengar et al [20]. advised an entropy-based method to pact with examined VoIP packets, over-load identification, and diminish server congestion. They utilized the Hellinger distance to select the threshold value, by distinguishing current congestion and standard behavior. Nevertheless, this method was regarded as Quick identification and distinction between the assailant and legal users but as well, some characteristics have been absent. Individual SIP proxy servers and ADS badly damaged the system congestion and overloading the performance. Superior QoS was not provided for time-sensitive services like that as IP Phone... etc. This method added to the acknowledgment of unusual congestion, nevertheless, it came up briefly in a specific context.

3CX Phone System [21] is an item-based IP Private Branch Exchange (PBX) that replaces a standard PBX and passes on agents the ability to make, get and move calls. The proxy server has a catalog, overall/clients, and their corresponding SIP address with the help of the registrar server and hence can connect an internal call or route an external call using the VOIP gateway. 3CX has numerous deformities if it was installed on individual SIP servers, for example, Failover to some degree restricted where the Setting up failover is relatively complicated and fairly limited in performance; it is not steady if the amount of the users has been extended; It crashes and call dropping and it is critical to restarting it; 3CX offers very little customization outside of the UI, and normally telephone framework supervisors will be essentially incapable to perform more-complex call routing; dynamic outbound routes are just unrealistic outside of the overly-simple Outbound Routes tab, and agents are limited to just prefix, length of destination, and from extension; SIP service (SRV) records are broken and not agreeable with the SIP RFC.

2. Proposed Framework

2.1.Problem statement

Installing scalable VoIP implies expanding the clients through the network, anyway adequate business, service, network, and component the board devices and cycles are not accessible. One of the difficulties seems when the current VM needs more resources to proceed with arriving calls and placing the calls into a queue, waiting for available resources. We solved this challenge by designing and installing a Sophisticated Cloud phone on the private Cloud of the Main data center of the Egyptian ministry of interior EMOI to foster structure and empowering intranet calls frameworks among various sectors, and directorates. Another challenge appeared that the proposed framework needs to be protected against probable attacks and hackers particularly Since SIP is utilized on the public Internet. A massive need for the provision to design a model to detect these assaults, especially DDoS attacks. Besides, the Quality-of-service necessity of VoIP does not limit the processing time of packets by these security measures remarkable limit.

2.2.Framework Overview

The initial structure of our proposed cloud phone on the Private Cloud of EMOI depends on the utilization of a few clustered SIP servers as demonstrated in Fig 4. It utilizes a 3CX cloud phone consists of eight nodes each has a 3CX running process with the unique IP address that is utilized by end clients to an interface. See Fig. 5

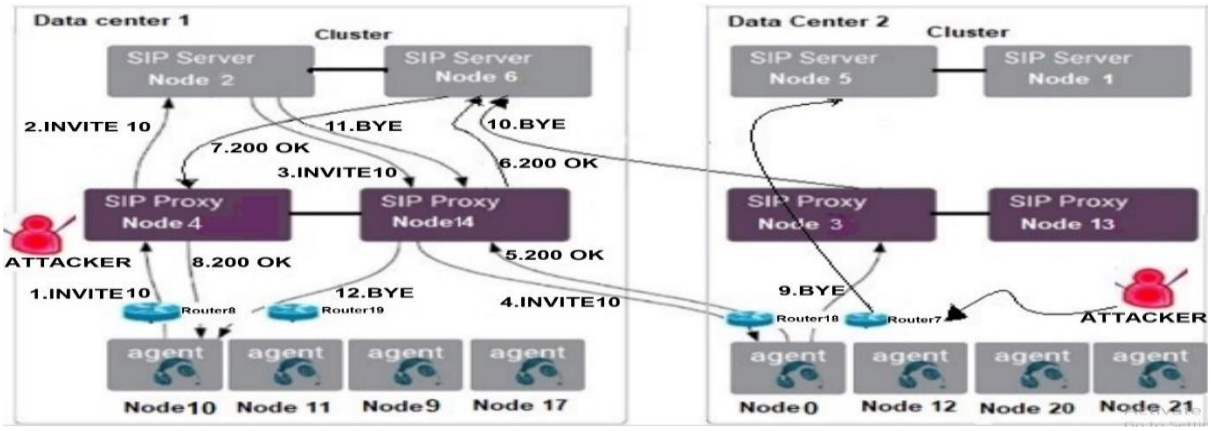
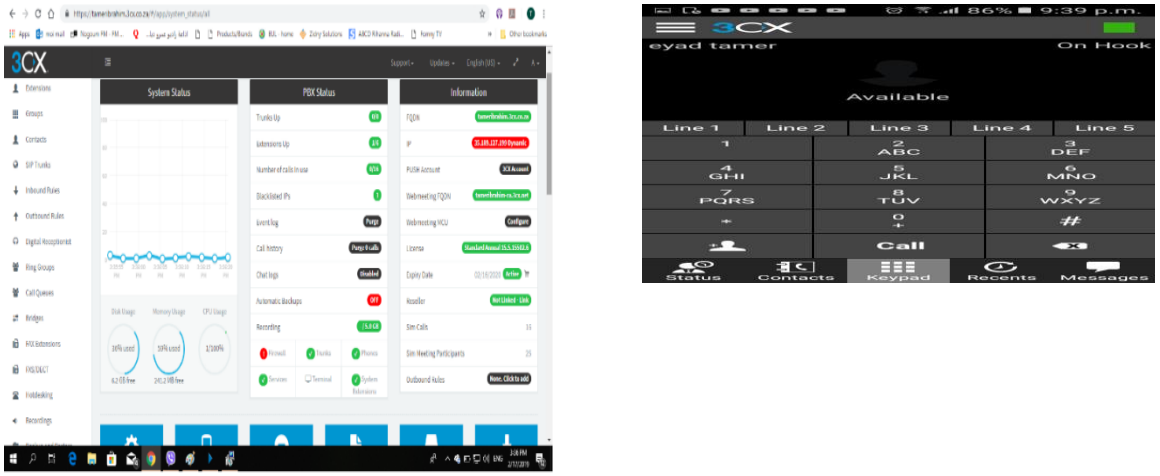


Fig.4: Proposed framework infrastructure and components



a) 3CX Server home page **b) 3CX web client (smart phone)**

Fig. 5: Proposed Cloud Phone application interface

Our proposed system architecture consists of three stages: In the first stage, before any User-agent requests to start communication with another client, each client ought to enroll with a Registrar server. Node number 10 beginnings the connection with Node number 0 by sending INVITE Request. The connection request shows up at the SIP proxy server Node 4 from Node 10, which contains the logical SIP address to the target N10.SIP server Node 4 demands the information from the SIP Location server Node 2 for the physical area of the goal. The proxy server settles the resulting address into an IP address from the DNS SIP server Node 6. The proxy server Node 14 sends a request to the callee using the target IP address. The callee responds to the SIP proxy server Node 14 with a response determining if the request is recognized and sends its response back to the Proxy server, who along these lines passes this response back to the visitor. The visitor develops a direct RTP connection with the callee and starts a connection with it. The connection request of the visitor arrives at the proxy server of its space, containing the target SIP address of the callee. Essentially, when user-agent have begun a session with one another, they become user-agent customers and user-agent servers to one another and can invite additional individuals into the session. In The second stage, the assailant attempts to enroll himself as the targeted VoIP client. Every one of the incoming calls to the victim VoIP client will be routed to the VoIP telephone of the aggressor rather than the victim's VoIP telephone. The proposed outline figures the traffic limit to estimate if there is strange traffic or not for identifying these assaults. In the third stage, the proxy-clustered nodes were denied admittance to authentic packets due to spamming of DDOS assault packets. At that point, real packets are dropped because they failed to arrive at the objective SIP server.

2.3.Framework architecture

The implementation of our Cloud phone framework was checked and tested through CLOUDSIM and NS2. It consists of 23 nodes, which can be expanded further as demonstrated in fig 4. The vital function of these nodes is clarified as follows: Clustered SIP servers: Node 1, 2, 5, and 6. act as centralized dynamic load balancers. The victim is node 5, and ADS (anomalies detection system) is node 3 Clustered Proxy servers: Node 3, 4, 13, and 14. SIP UA: Node: 0, 9, 10, 11, 12, 17, 20, and 21. DDOS Attacker: Node 15, 16. Attack Packets: Red-colored Packets Legitimate with INVITE flood traffic generator situated in a remote location to destruct network resources. Legal Packets: Black-colored Packets. Dummy Destination: Node 22 - This is to redirect the attack packets reaching the server. Routers: Node 8, 7, 18, and 19 with wide area network emulator by installing NIST Net. Our proposed framework computes normal network conduct utilizing standard Hellinger distance equations to decide threshold value by observing the next period. The threshold value is a considered performance measure because the small values of this parameter helps in quick assaults detection. If there is any deviation in normal conduct or over-load, our module will recognize it. Our method

compares the current following conduct to the normal conduct and packet assault detection. With an INVITE alarm, a DDOS assault is recognized if the protocol conduct distance surpasses the threshold value, see Fig. 6.

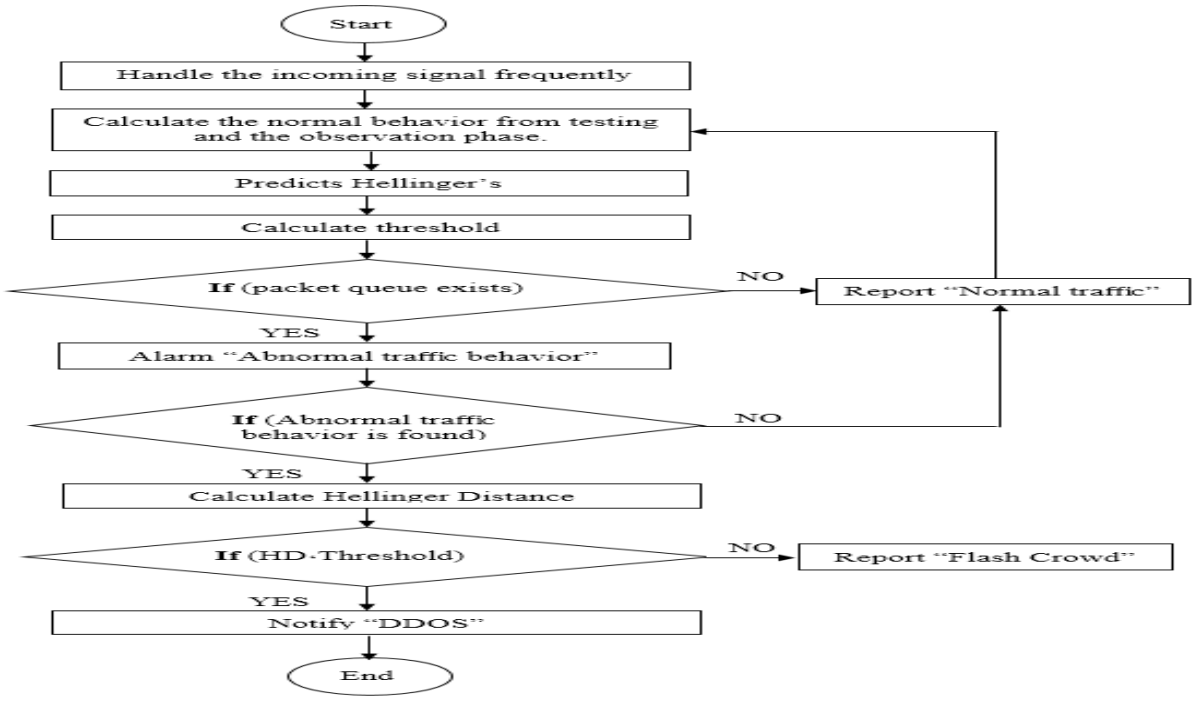


Fig. 6: Flowchart of the proposed software implementation

3. INSTALLATION AND SIMULATION

3.1.Implementation

Our framework is based on the usage of the Ubuntu Linux server as contextual analysis. The server was installed with Ubuntu Linux 14.04 operating system with NS2 version 6 and CloudSim 3.0.3 for testing and simulation. A wide area network emulator (NIST Net) was installed on a Linux server to create virtual routers connected to enterprise networks and SIP servers. CloudSim package was installed, unpacked, and add it to the Java class-path and it is fit to be used.

3.2.Framework Simulation

There are two languages are utilized for NS2, TCL tool command language as front end and C++ as back end, and one language is utilized for CloudSim, Java as back end. Also, writes in TCL script. They are interpreted by a network simulator and give two output files are NAM and TR files. NAM is for visual animation of the output and TR is the enormous text trace file comprised of the simulation results.

3.2.1 Call Allocation Simulation

CPU-utilization procedures were examined first. Second, time-awareness procedures were simulated when these procedures consider the specific time of VM provisioning. Finally, load-aware adaptations were examined when methodologies think about loads of VMs and their varieties. Eventually, a plan of another strategy was contributed to improving the performance by limiting the call delay and the billing hour's expenses as considered performance measures. The least call delay positively affects the QoS overall system. On the side, Small numbers of billing hours aid in improving call allocation and mitigate its waiting in the queue. In order to assess their robustness, we consolidate to all techniques eight StUp delays: 0, 45, 90, 135, 180, 225, 270, and 315 seconds as experiments. To accomplish best load balancing, consolidating three fundamental techniques: Utilization awareness (UA), Time awareness (TA), and Load awareness (LA) were combined and applied as opposed to applying just a

single strategy as past ways to fulfill better performance. This method contributed to improving designations of VoIP calls by assigning job j to VM with nearest processing finish time, which has the least CPU usage left that utilizes the minimal periods of x seconds to measure future load. Best outcomes were accomplished when contrasted with the five procedures Ffit (Allocate job to the first VM capable to execute it), Bfit (Allocate job to VM with the smallest utilization left), Wfit (Allocate job to VM with the largest utilization left), Rand (Allocate job to VM randomly), and RR (Allocate job to VM using Round Robin).

First, to examine CPU Utilization awareness, we assessed the performance of five strategies compared to our proposed technique. Six workloads were analyzed and simulated; each incorporates phone calls made during one day in six distinct days with various workloads every day. The results showed that our technique uses about 10.5 billing hours, while Rand, RR, and WFit use about 17 billing hours, Other outcomes showed that our framework has the least billing hours 16.2 billing hours on average, on the other side Rand strategy had the worst strategy equal to 22.5 billing hours on average. See Figs.7, 8.

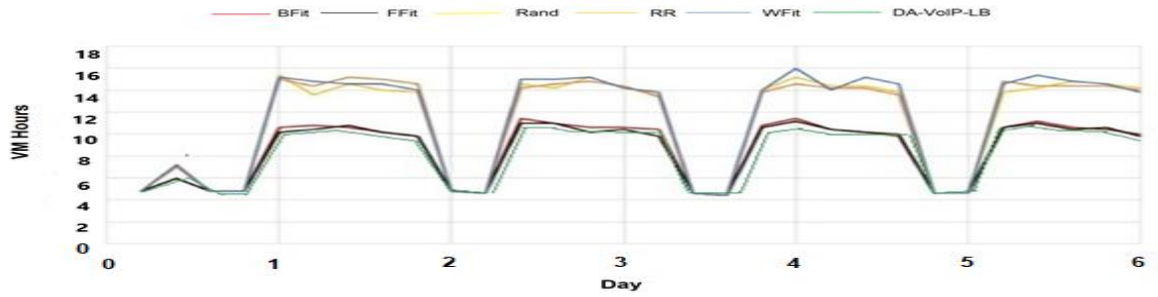


Fig. 7: the number of billing hours during six days (UA Strategy)

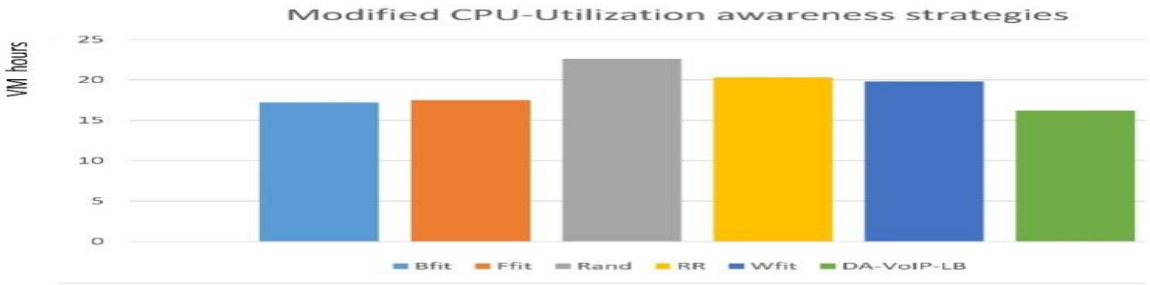


Fig. 8: Average billing hours per day (UA Strategy)

Fig. 9 shows an illustration of the number of billing hours during a day. The greatest number of VMs running during top hours is 4 VMs only, which is lower than past approaches.

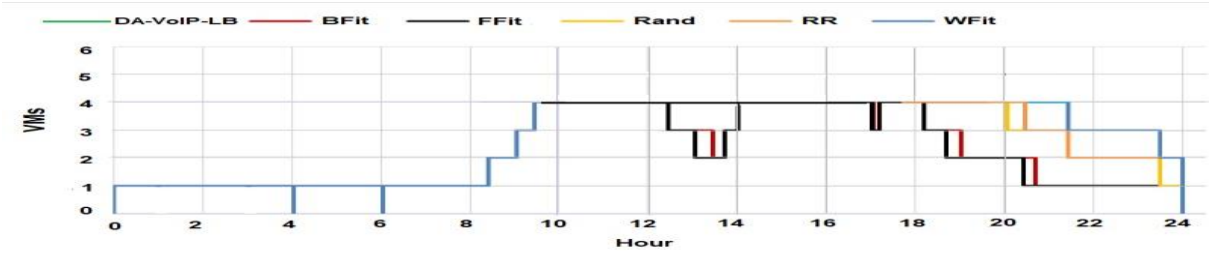


Fig9: Illustration of the number of billing hours during a day (UA Strategy)

Second, to examine the Time-aware strategy, we conducted experiments including twenty-three techniques furthermore, our proposed framework to be assessed by using cloudsim: BFit, FFit, MaxFTFit, MidFTFit, MinFTFit, Rand, and RR, and WFit, three time-aware versions of BFit, FFit,

Rand, RR, and WFit. Fig. 10 shows the Billing Hours (b'), degradation versus StUps. It was noticed that our methodology achieved the best performance compared to the other techniques. The most noticeably terrible techniques are MinFTFit and WFit. Fig 11 demonstrates that the best average Calls to Queue (c') is about 2 in a day during 6 days for our technique DA-VoIP-LB, and on the other side the average calls for MaxFTFit in a day are about 4 for 6 days (the worst strategy) with StUp equals 315 sec. Our proposed technique aims to improve the performance by limiting the daily average number of billing hours and mitigating the number of holding up calls in the queue.

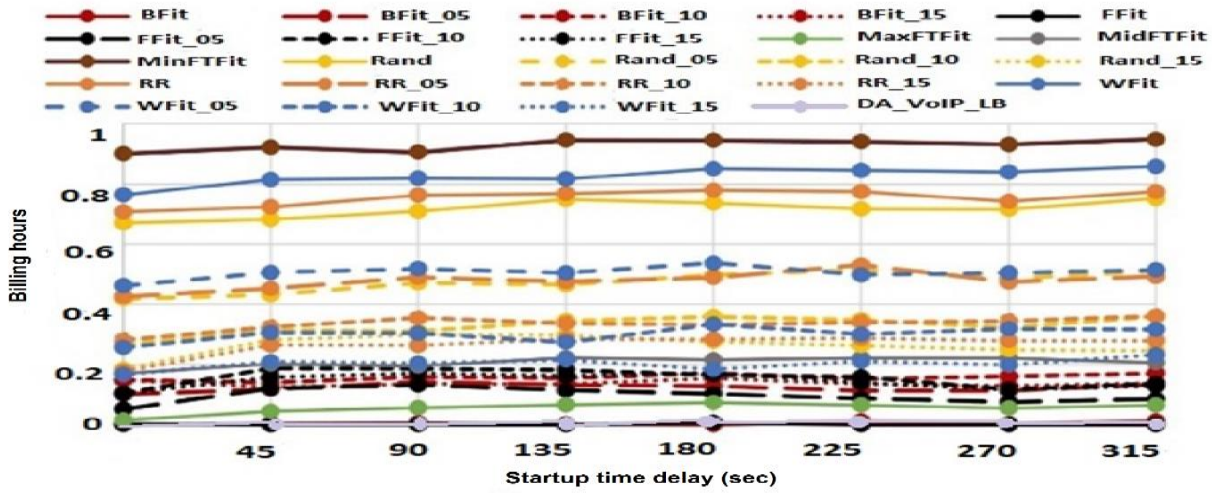


Fig. 10: billing hour's degradation (TA Strategy)

Third, to examine the Load-aware strategy, Twenty-four procedures with four predictions were assessed and tested, time intervals each 10, 20, 30, and StUp (time interval is equivalent to startup time delay). Our proposed method with four predictions was compared to four previous methods BFit, WFit, RR, and Rand. It was seen that our methodology gave a better performance with average calls to Queue (c') is around 1 per day, and the most noticeably awful are WFit_s10, WFit_s20, and WFit_s30.

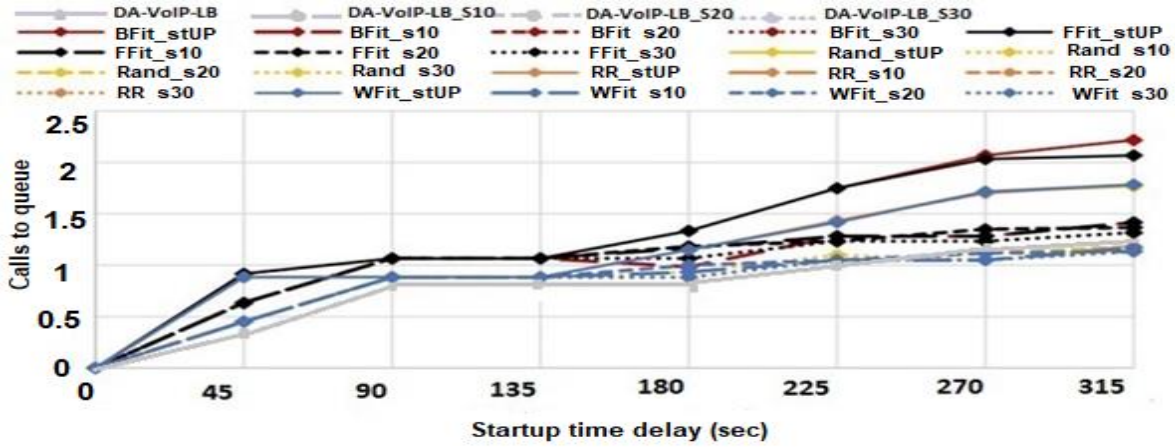


Fig. 11: average calls to a queue (UA Strategy)

The average Calls to Queue (c') is around 2.5 per day for the worst methodology BFit_stUp with StUp approaches 315 sec, see Fig. 12. Fig. 13 means the billing hour’s degradation.

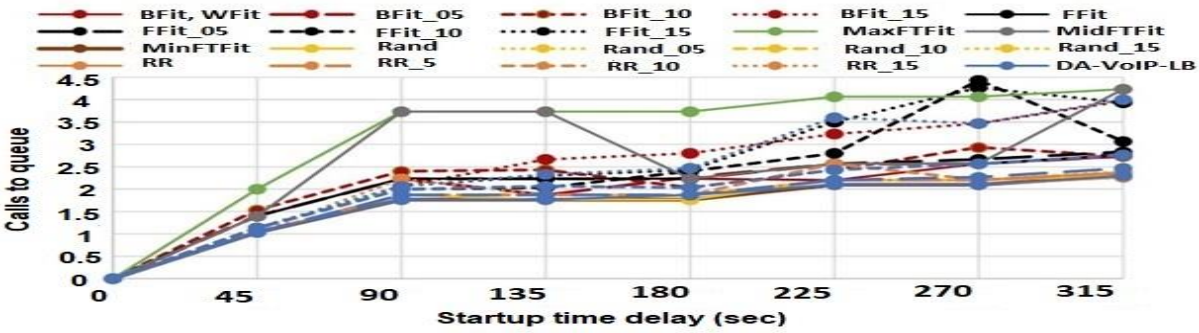


Fig. 12: average calls to queue (LA Strategy)

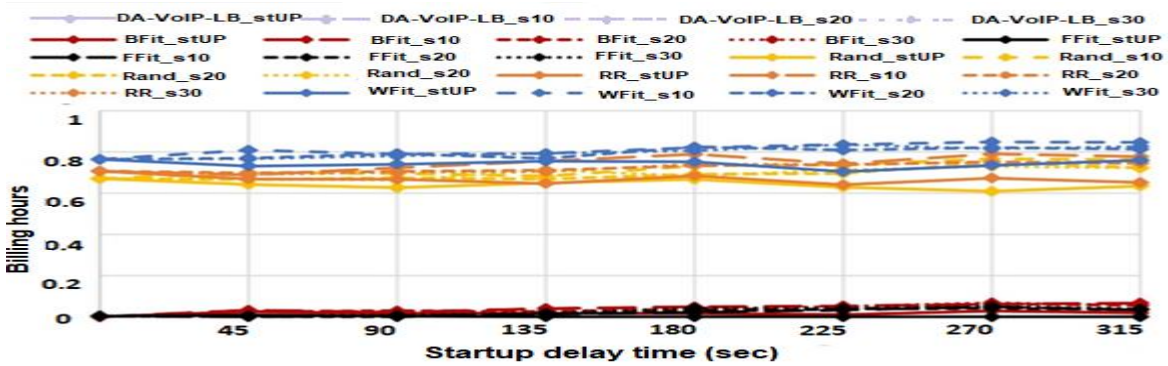


Fig. 13: billing hour's degradation (LA Strategy)

In table 1, Position b' depends on the billing hours degradation. Position c' alludes to the position concerning the calls to the queue. The position is the position-dependent on the averaging two rankings. When our methodology was applied to dispense job j to VM, which has the nearest complete time, and least CPU usage with 10, 20, and 30-sec intervals to anticipate the load, the outcomes were superior to past outcomes when compared with the referenced five strategies.

Table 1: Comparison between degradation parameters of DA-VoIP-LB and BFit

Strategy	Deg_ b'	Deg_ c'
BFit	0.0214	0.1642
BFit_stUp	0.0073	0.5547
BFit_s10	0.0190	0.1542
BFit_s20	0.0160	0.1791
DA-VoIP-LB	0.0080	0.0009

4.2.2 Simulation For Protecting Proposed Framework

VoIP packets are produced at the client-side following a Poisson distribution i.e. the packets follow exponential arrival rate on the servers, the packets have varying lengths and the lengths are simulated by the exponential distribution. Latency (propagation delay) was set to 100 MS per link, 0.5Mbps link speed in a fixed Bandwidth, and the packet loss rate to 0.42% in our tests as default values. Each link has the same length and each link offers an equivalent service. Each link is using 100 Mbps Ethernet. Python code was set up through the environment of the proposed framework, pass some of the vital capacities, and run the simulation. NS2 was utilized to create VoIP traffic before any unusual traffic. It

was deduced in some of the past approaches that ADS (Anomaly Detection sensor) is time-consuming to analyze both incoming and normal traffic profiles, which harm the performance. ADS was installed on the same SIP 4 proxy node to diminish time and network traffic utilization, utilize network resources, and accomplish high accessibility. ADS monitors both tracking of signals between UAs and the SIP proxy cluster nodes.

The initial five minutes show normal behavior and toward the beginning of the fifth minute, there is a horde of call demands. Note that toward the finish of 5.40th min, the distance starts to climb and arrives at a maximum of 0.043 at 5.60th min. At a higher rate, the dropping or retransmission of a few packets is covered up and does not bring about any noticeable deviation. During the INVITE alert, if the protocol behavior distance stays less than the threshold value of ($\mu + 8 \times \sigma = 1.841\times10-3+8\times(5.40\times10-3) \approx 43\times10-3$) at that point it is a flash crowd event. Toward the beginning of the eighth minute of the simulation, the spoofed source IP node assaults the SIP servers. SIP proxy cluster moves some INVITE messages of the accepted spoofed INVITE requests toward the objective IP address. When the flood rate expands, most of the resources of the node are consumed through spoofed demands. Subsequently, SIP servers' performance degrades once more, demonstrating unstable recuperation and debasement conduct. The server was denied admittance to authentic black packets because of spamming of RED assault packets. At that point, real packets are dropped due to their inability to arrive at the server as demonstrated in Fig. 14.

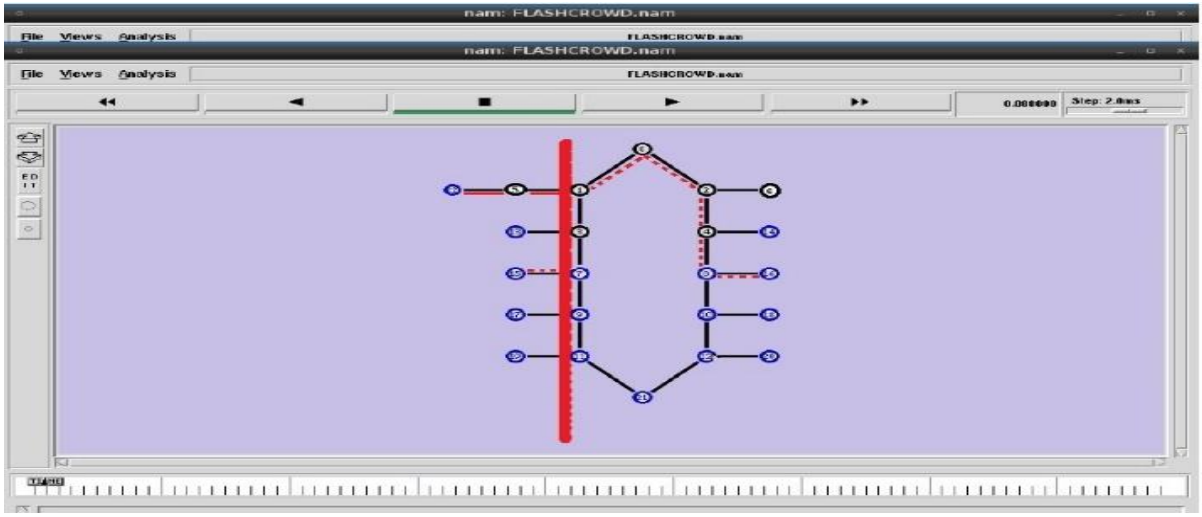


Fig. 14: **DDOS causes severe flooding at server (i.e., server resource unavailable even for legitimate users**)

4. RESULTS

This paper presents a new VoIP framework over EMOI Private Cloud. New procedures were taken for improving the allocation of the call to achieve scalability. Other techniques were conducted to secure this framework by detecting DDOS attacks. Scheduling problems of tending cloud-based VoIP load-aware scheduling methodologies with VM startup expectations were formed and considered. Billing hours were utilized as a parameter to quantify how the calls were balanced on VMs. The outcomes showed that our proposed approach has an excellent service putting less than 0.1% of calls to the holding up queue all out during six days with the minimum percentage of billing hours in the most pessimistic scenario, which is viewed as the best outcome contrasted and comparative techniques. In our framework, combining the techniques of load awareness, CPU utilization, and time awareness is a key performance metric for VoIP quality of service estimation. It tends to be utilized to follow QoS relapses when it increments above a specific threshold, or improvement, when it is below and is useful for VoIP QoS issue examining. It was demonstrated that due to these strategies, the performance is changing over time, allotment adjusts to these changes. Table 2 shows that our proposal sees more subtleties on utilizing Private Cloud phones with enough clustered redundant nodes to accomplish high availability and scalability.

Table 2: Comparison between SDA-VoIP-LB and similar approaches

Parameter	Our framework	Past approaches
Scalability	Yes	No
Internet delay	50 MS	50 MS
Packet loss rate	0.42%	0.42%
Rate of server's degradation	Low	High
Rate of server's degradation	Slow	Quick
Time to reach maximum of flash crowd	At 5.6^{th} min	At 6.33^{th} min
Time to start spoofed invite request	At 2.5^{nd} min	At 3.1^{rd} min
Time to reach Top DDOS	At 2.7^{nd} min	At 3.4^{rd} min
Maximum of DDOS distance	0.125	0.135
Packet loss during DDoS occurrence	41%	50%

SUMMARY AND CONCLUSIONS

In this paper, we assessed the performance of the proposed cloud phone installed on the private Cloud of the Egyptian Ministry of Interior EMOI. The system performed under the 3CX SIP VoIP Background, and Cloudsim and NS2 as Model simulation, taking into examination various appraisal standards: call installation delay, full delay, MOS scoring, throughput, and queue delay. This paper breaks down the highlights of our techniques to improve the detection of DDOS. Our technique depends on the utilization of Hellinger Distance to calculate the threshold. Our framework comprises user agents, routers, clustered nodes that work as Proxy, registrar, and location SIP servers, which accomplishes high availability, quicker, better viable detection of DDOS floods, mitigation of risks emerging from malignant assaults, improve connection rates, and limit the delay. It was deducted and demonstrated in the wake of finishing the analyses that the number of clients of that model at whatever point they extended don't contrarily influence the performance when the proposed call distributions methodologies would be applied through the SIP Proxy clustered servers. Scalability was accomplished in all pieces of the proposed structure, which is important and significant in such these created frameworks particularly in these imperative areas to give quick and secured support to the clients. Nevertheless, further investigation is needed to assess its genuine productivity and effectiveness in every area. Moreover, dynamic consolidation and load balancing is another significant issue to be addressed.

ACKNOWLEDGMENTS

The authors wish to acknowledge the support of the Systems and Computers Engineering Department, Faculty of Engineering, Al Azhar University, Cairo, Egypt.

REFERENCES

[1] Albeiro Patiño Builes (2015) Technology Trends for Business Productivity Increase. INGE CUC vol. 11 no. 2, pp. 84-96. Barranquilla. ISSN 0122-6517 Printed, ISSN 2382-4700 Online.

[2] Eric Simmon (2018) Evaluation of Cloud Computing Services Based on NIST SP 800-145. NIST Special Publication 500-322, NIST Cloud Computing Cloud Services Working Group NIST Cloud Computing Program Information Technology Laboratory.

[3] Darshan Lal Valecha (2015) Intelligence Taxation with Cloud Governance. International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online).

[4] Nitin Kumar Mishra, Nishchol Mishra (2015), Load Balancing Techniques: Need, Objectives and Major Challenges in Cloud Computing- A Systematic Review. International Journal of Computer Applications (0975 – 8887) Volume 131 – No.18.

[5] Geetha Megharaj, Dr. Mohan K.G.2 (2016), A Survey on Load Balancing Techniques in Cloud Computing. IOSR Journal of Computer Engineering (IOSR-JCE). E-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 18, Issue 2, Ver. I.

[6] PAWAN KUMAR, RAKESH KUMAR (2019), issues and Challenges of Load Balancing Techniques in Cloud Computing: A Survey. ACM Computing Surveys, DOI: 10.1145/3281010.

[7] SOURABH SHASTRI, 2ARQUM HAMID, 3VIBHAKAR MANSOTRA (2017), SOURABH SHASTRI, 2ARQUM HAMID, 3VIBHAKAR MANSOTRA. International Journal of Advanced Computational Engineering and Networking, ISSN: 2320-2106, Volume-5, Issue-10.

[8] Ali H. Wheeb (2017), Performance Analysis of VoIP in Wireless Networks. IRACST – International Journal of Computer Networks and Wireless Communications (IJCNWC), ISSN: 2250-3501 Vol.7, No 4.

[9] Jorge M. Cortés-Mendoza, Andrei Tchernykh, Alexander Yu. Drozdov, Pascal Bouvry, Ana-Maria Simionovici, Arutyun Avetisyan (2015), Distributed Adaptive VoIP Load Balancing in Hybrid Clouds. Ministry of Education and Science of Russian Federation under contract, No02.G25.31.0061.

[10] Nikita Haryani, Dhanamma Jagli (2014), Dynamic Method for Load Balancing in Cloud Computing. IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 16, Issue 4, Ver. IV.

[11] Prof Anita M. Pujar, Pooja S. Kshirsagar (2017), Resource Allocation Strategy with Lease Policy and Dynamic Load Balancing. I.J. Modern Education and Computer Science, 2017, 2, 27-33 Published Online February 2017 in MECS (http://www.mecs-press.org/) DOI: 10.5815/ijmecs.

[12] Sunny Nandwani, Mohit Achhra, Raveena Shah, Sowmiya Raksha (2016) , Weight-Based Data Center Selection Algorithm in Cloud Computing Environment. Artificial Intelligence and Evolutionary Computations in Engineering Systems, DOI: 10.1007/978-81-322-2656-7_47.

[13] Venkateshwarlu Velde∗ and B. Rama (2019), Optimized Weighted Round Robin Approach for Load Balancing in Cloud Computing. Journal of Computational and Theoretical Nanoscience Vol. 16, 1–8.

[14] Mohammad Alhihi, Mohammad Reza Khosravi, Hani Attar, Mohammad Samour (2013), Determining the Optimum Number of Paths for Realization of Multi-path Routing in MPLS-TE Networks. Determining the Optimum Number of Paths for Realization of Multi-path Routing in MPLS-TE Networks.

[15] Faritha Banu, K. G. Shanthi, P. Lakshmi Priya, M. Faritha Begum (2016), Performance Enhancement Architecture of VoIP Applications in Multiprotocol Label Switching Networks. Circuits and Systems, 2016, 7, 2047-2058.

[16] Jorge M. Cortés-Mendoza1, Andrei Tchernykh, Igor Bychkov, Alexander Feoktistov, Pascal Bouvry (2017), Load-Aware Strategies for Cloud-based VoIP Optimization with VM Startup Prediction. 2017 IEEE International Parallel and Distributed Processing Symposium Workshops, 978-0-7695-6149-3/17 $31.00 © 2017 IEEE DOI 10.1109/IPDPSW.2017.73.

[17] Mosleh M. Abualhaj, Mayy M. Al-Tahrawi2, Sumaya N. Al-Khatib3 (2019), Performance Evaluation of Voip Systems in Cloud Computing. Journal of Engineering Science and Technology Vol. 14, No. 3 (2019) 1398 – 1405.

[18] Manjur Kolhar, Abdalla Alameen, Mujthaba Gulam (2018), Performance evaluation of framework of VoIP/SIP server under virtualization environment along with the most common security threats. Neural Comput & Applic (2018) 30:2873–2881.

[19] Amita Chauhan, Nitish Mahajan, Harish Kumar, Sakshi Kaushal, (2019). Analysis of DDoS Attacks in Heterogeneous VoIP Networks. International Journal of Innovative Technology and Exploring Engineering (IJITEE), a Survey, ISSN: 2278-3075, Volume-8, Issue-6S3, April 2019.

[20] N. Jeyanthi and N. Ch. Sriman Narayana Iyengar, (2012). An Entropy Based Approach to Detect and Distinguish DDoS Attacks from Flash Crowds in VoIP Networks, International Journal of Network Security, Vol.14, No.5, PP.257-269, Sept. 2012.

[21] Manual 3CX Phone System for Windows Version 10.0.