Implementation of scalable VoIP System over Cloud

Tamer Elnawawy1, Islam Hany Harb2

Abstract - The tremendous growth of internet and cloud computing systems puts enormous pressure on internet service providers to ensure quality of service for real time applications, such as voice over IP (VoIP) and video on demand. VoIP leverages the use of packets, therefore, much more information can be carried over the network to support/enhance our communication needs, compared to traditional telephony methods. This introduces scalability and reliability challenges to handle such massive data exchange over the internet. In addition, virtualization, in cloud computing, adds another dimension to the problem: distribution of the virtual machines (VMs) and their migration. The repeated use of Internet Protocol shortest path towards the same destination may lead to unbalanced traffic situations and degraded network performance. Therefore, load balancing and link utilization became critical functions in Internet Protocol routing to provide Quality of Service (QoS) assurance for VoIP applications. In this paper, a study of the most popular approaches and techniques of VoIP load balancing over cloud is conducted and our BFit approach is proposed to enhance the quality of service for VoIP applications. The proposed approach is implemented with an effective flow classification technique. It prioritizes the voice packets based on their flow arrival rate and bandwidth utilization. A comprehensive model for VoIP load balancing is implemented. To evaluate our approach, an integrated VoIP system is installed using 3CX over Cloud. NS2, a simulation tool, is used to generate VoIP traffic and apply different Load balancing approaches.

Index terms—Cloud, VoIP, Load balancing, 3CX, SIP, MPLS, QoS, delay, jitter, bandwidth, packet loss, and throughput

1 INTRODUCTION

VOIP is a group of technologies and a methodology for delivery of voice and multimedia over Internet Protocol (IP) networks which all sorts of networks support: corporate, private, public, cable, and even wireless networks [1]. It means voice transmitted over a digital network. It is also commonly referred to as broadband telephony, IP telephony, Internet telephony, and broadband phone service. Internet is not strictly necessary for VoIP but what is necessary is the use of the same protocols that the Internet uses. VoIP technology enables traditional telephony services to operate over computer networks using packet-switched protocols. Packet-switched VoIP puts voice signals into packets, similar to an electronic envelope. Along with the voice signals, the VoIP packet includes both the caller's and the receiver's network addresses. VoIP packets can traverse any VoIP-compatible network.

TAMER ELNAWAW is the head of project/management section, IT Department, Ministry of Interior, Cairo, Egypt. Assistant, Al-Azhar University, Faculty of Engineering, Computers & Systems Dept. MSC of Computers & Systems Engineering with Excellent Degree, Al-Azhar University 2014. BSC of Electrical Engineering, Automated Control, and Computer Systems with Good Degree, Tanta University 2000, PH:00201023177787 (E-mail: tamer.nawawy@gmail.com),Cairo, Egypt

ISLAM HARB is an engineer in Mentor Graphics - a Siemens Business, USA. Ph.D. in Computers & Systems Engineering, Al-Azhar University, Egypt. MSc. in Computer Science, College of Engineering, Virginia Tech, USA. BSc. in Computer Engineering, Cairo University, Egypt

The main challenge of VoIP applications over Cloud become to keep the performance same or better whenever such an outburst occurs [2]. One of these challenges and problems which has to be solved is Load balancing. It is considered as one of the prerequisites to utilize the full resources of parallel and distributed systems. The load balancing in the cloud is also referred as load balancing as a service (LBaaS). It moves the traffic from congested links to alternative paths in the network. It is a process to make effective resource utilization by reassigning the total load to the individual nodes of the collective system and to improve the response time of the job.

It distributes the dynamic workload across multiple nodes to ensure that no single resource is either overwhelmed or underutilized. Load Balancing is done with the help of Data Center Controllers, which uses A VmLoadBalancer to determine which VM should be assigned the next request for Processing. Based on predetermined parameters, such as current load and availability, the load balancer uses various scheduling algorithms to determine which server should handle and forward the request on to the selected server.

A) Load balancing types:

On the bases of the current state of the system, load-balancing algorithm can be categories into two types [3]:

1- Static approach: -

This approach is mainly defined in the design or implementation of the system. Prior knowledge about the system is already known which includes processing power, memory, performance and data about user's requirements. These algorithms do not need the information regarding current state of the system. Static load balancing algorithms divide the traffic equivalently between all servers. By this approach, the traffic on the servers will be disdained easily and

consequently it will make the situation more imperfectly. Many techniques use this approach such as Round Robin, Min-Min, Max-Min, Opportunistic, Throttled, and FCFS algorithms.

2- Dynamic approach: -

This approach considered only the current state of the system during load balancing decisions. It is more suitable for widely distributed systems such as cloud computing. Designated proper weights on servers and by searching in whole network a lightest server preferred to balance the traffic. However, selecting an appropriate server needed real time communication with the networks, which will lead to extra traffic added on system. Stochastic Hill climbing, Meta-Heuristics, GA, Event-driven, Honeybee, Ant Colony...Etc. are examples of this approach.

In comparison between static approach and dynamic approach, we found that although round robin algorithms based on simple rule, more loads conceived on servers and thus imbalanced traffic discovered as a result. However; dynamic algorithm predicated on query that can be made frequently on servers, but sometimes prevailed traffic will prevent these queries to be answered, and correspondingly more added overhead can be distinguished on network.

B) Cloud based VoIP Features:

A cloud based VoIP can further reduce costs, add new features and capabilities, provide easier implementations, and integrate services that are dynamically scalable. Other benefits include data transfer availability, integrity, and security [3]. VoIP requires the service availability all the time for any number of users. To deal with increasing number of clients, providers can invest in a large infrastructure to avoid loss of calls (hence, users). VoIP calls require signaling, channel setup, voice signal digitization, encoding, etc. The voice nodes handle the calls with different features such as: voicemail, call forwarding, music on hold, conference calls, etc. depending on customers. In a cloud-based VoIP solution, the voice nodes are operated as VMs that provide a variety of services. Distributed cloud based VoIP architecture assumes that voice nodes are distributed geographically; hence, they are grouped in different locations (data centers). To deploy and effectively manage telephones via clouds different characteristics need to be improved. The most important is the utilization of the infrastructure.

C) VoIP Mechanism:

The audio signal is transformed into digital form by an analog-to-digital converter and voice data is packetized and encoded prior to transmission of the signal at the sender is shown in figure.1. Encoding - decoding is done by Coder-decoder (CODEC) that transform sampled voice data into a specific network-level representation and vice versa. Most of the codecs are defined by standards of the International Telecommunication Union, the Telecommunication division (ITU-T). Each of them has different properties regarding the amount of bandwidth, encoding standards and delay it requires. A common benchmark used to determine the quality of voice is the Mean Opinion Score (MOS). It evaluates the quality of speech provided by a codec. Each codec provides a certain quality of speech only if processor utilization is low enough. Montazerolghaem et al. [4] reported that the consumed bandwidth of 6,500 calls per second not exceed 100 Mbps, and 10,000 calls not reach 400 Mbps. Table I shows the bandwidth used by different codecs considering that VoIP calls use audio streams for endpoints (a call between two parties will use double of bandwidth). The number of calls supported in 100 Mbps connection is between 6,000 and 25,000 depending on codec of the calls.

Fig. 1 VoIP System

Fig.1. VoIP System

Receiver encapsulates and decodes the received VoIP packets. Decoding may include dejittering, error correction and packet loss concealment. The digital data is then converted to analogue form again and played at an output device.

TABLE 1
ITU standards CODEC types

Codec	Bit Rate (kbps)	MOS	Bandwidth (kbps)
G.711	64	4.1	87.2
G.729	8	3.92	31.2
G.723.1	6.3	3.9	21.9
G.723.1	5.3	3.8	20.8
G.726	32	3.85	55.2
G.726	24	-	47.2
G.728	16	3.61	31.5
G722_64k	64	4.13	87.2
ilbc_mode_20	15.2	NA	38.4
ilbc_mode_30	13.33	NA	28.8

The three main factors that mainly affect VoIP quality are delay, jitter, and packet loss. Most users recognize roundtrip delays when they exceed 250 Ms. Network congestion, timing drift, or route changes can cause Jitter and may differ for each packet [5].

1-Latency (delay) is the mouth-to-ear overall delay. According to [25], the delay has five different components:

Encoding delay: It is the time interval needed to encode the voice signal, it is defined by the voice codec.

Packetization delay: It is the time interval needed to packetize the voice frames.

Network delay: It is the combination of transmission, propagation and queuing delays.

Playback delay: It is the delay caused by the playback buffer of the receiver's side, it helps to smooth jitter metric of voice packets.

Decoding delay: It is the time interval needed for reconstructing the voice.

2-Jitter can be defined as one-way delay variation and influences quality if it exceeds a maximum value.

3-Packet loss occurs when packets sent are not properly received for playback by the other end. Overloaded links, buffer size in the receiving device, collisions, congestions in link, physical layer errors etc., can cause packet losses. When the packet losses arises the receiver either introduces gaps in playback or tries to recover from this error by using Packet loss concealment, forward error correction techniques.

Bandwidth defines the size of the link capacity. Streaming videos, or making uninterrupted video calls need a large bandwidth, therefore it should be optimized in an effective way. It is stated in [5] that, "Several compression/decompression (codec) algorithms recommended by the ITU-T can reduce the amount of bandwidth needed for one VoIP circuit to a fraction of the traditional 64 Kbps of bandwidth reserved for calls in circuit-switched networks". Thus, using an appropriate codec and communication link play a big role on having a good voice quality.

D) VoIP Protocols:

The most commonly used protocols for transporting voice packets in IP networks are H.323, Session Initiation Protocol (SIP), Media Gateway Control Protocol (MGCP), H.248, RTP, RTCP, RTSP and RSVP.

1- H.323:

Is the ITU standard signaling protocol for establishing VoIP connections over packet switched networks.

2- SIP:

Is an Internet Engineering Task Force (IETF) standard text based open protocol for establishing VoIP connections. For call control, SIP uses the Session Description Protocol SDP, which describes the call details such as packet size, audio, or video stream and codec type. SIP is similar to H.323 and HTTP but specifically designed for the Internet to set up a session over the Internet.

3- MGCP:

Is the first protocol developed by the IETF to signal control information between VoIP network components. It is a master-slave protocol that allows a central coordinator (Call Agents) to monitor and instructs communication events and the gateways. They send media to specific addresses to establish and maintain VoIP communication path between endpoints and tear downs the path after communication.

2 BACKGROUND

A VoIP application is composed of several building blocks, as shown in fig. 2.

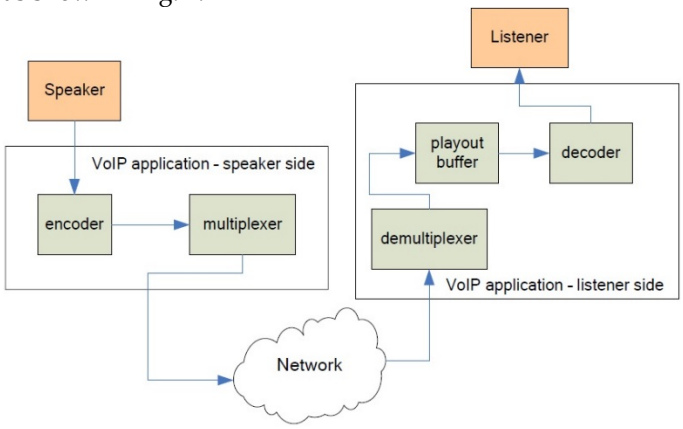


Fig 2. Scheme of a VoIP application

At the sender side, the first component is an encoder, which periodically samples the voice signal [7]. Encoders can be either sample based or frame based. The former, (e.g., G.711) code individual speech samples periodically the latter, instead, (e.g. G.729) group a certain number of samples within a time window (i.e. a frame) of some m milliseconds, and then code them together. For this reason, frame based encoders often achieve higher compression and smaller data rates than sample based ones, though at a higher encoding/decoding complexity. A number of speech frames can be multiplexed into the same packet payload, to reduce the overhead of transport, network and MAC headers, though at the expense of increasing the transmission delay.

At the receiver side, speech frames are de-multiplexed and inserted into a playout buffer [7]. A playout buffer enforces speech frames to be decoded at the same interval at which they were generated by the encoder. To do so, it might re-order, delay or even drop them if they arrive after their expected playback time. The playout buffer delivers speech frames to the decoder, which actually playbacks them. Some decoders implement packet loss concealment (PLC) techniques which allow missing speech frames to be somewhat reconstructed by interpolating (correctly received) surrounding frames. The conversation between the sender and the receiver can be in any of four states: A speaking, mutual silence, B speaking and double talk.

3 RELATED WORKS

Four main policies were discovered to present a full integrated scalable VoIP system on Cloud:

1) SIP dedicated load-balancing policy:

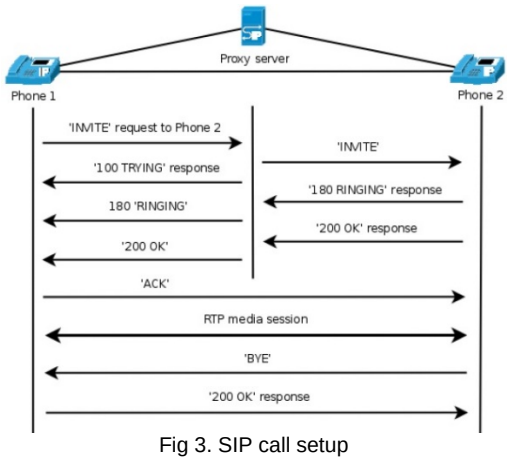


Fig 3. SIP call setup

The setup consists of two SIP phones using SIP proxy server to route calls to unknown destinations [8]. Phone 1 starts by sending and 'INVITE' request to phone 2 including SDP data for the requested session type. Since it doesn't know the IP address of phone 2 it sends the signal to the proxy server that sends a '100 trying' response back to phone 1 letting it know that the proxy server is trying to route the call. The proxy sends the 'INVITE' request from phone 1 to phone 2 that starts to ring, notifying the user of an incoming call. Phone 2 sends a '180 RINGING' response back to phone 1 through the proxy server and phone 1 also starts to ring, indicating that the call is being setup. When the user of phone 2 lifts the handset and answers the call the phone 2 sends a'200 OK' response to phone 1 via the proxy. The '200 OK' message contains the SDP media description that tells phone 1 what type of media session that phone 2 is able to handle for the call. Recall that in the initial 'INVITE' from phone 1 the session type request was sent thus the '200 OK' SDP media description completes a TWO-WAY negotiation of the capabilities that are to be used in the call. Phone 1 sends an 'ACK' message directly to phone 2 and the bidirectional RTP media stream is setup enabling phone 1 and 2's user to talk to each other. When the conversation is ended by any of the participants the terminating party's phone sends a 'BYE' request to the other part which responds with a '200 OK' message that terminates the RTP stream and the call is terminated.

When a SIP client requests to initiate a SIP call, SIP Load Balancer will perform a selection of the most appropriate SIP server, based: (a) on the DNS SRV records of the available SIP servers and (b) on various workload metrics collected from the existing SIP proxy servers, such as locations of the client and SIP server, SIP server load in terms of processing transactions, overall workload in each SIP server, etc. However, this solution is rather static as: (a) does not provide real time load metrics because as one or more SIP jobs finish in the proxies there is no way to inform back the LB and (b) in case a SIP proxy goes down e.g. due to hardware failure, the LB will continue to send it new transactions to dispatch. For the above reasons, we consider the frequent communication between the LB and SIP proxies as mandatory [8].

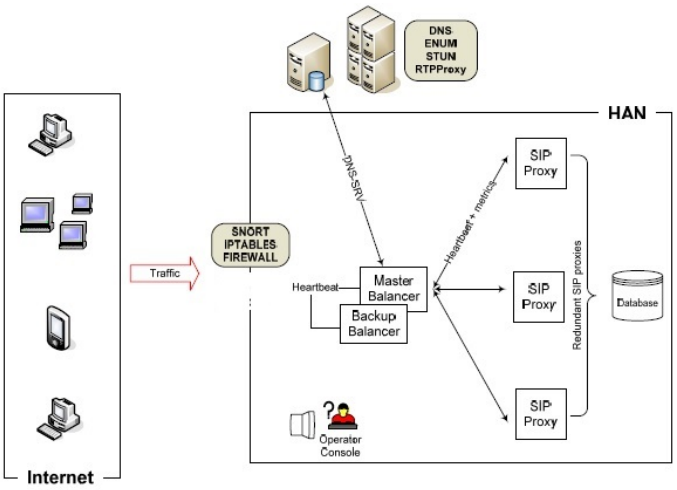


Fig 4. General high availability architecture for SIP domains

The LB is an add-on entity, which is responsible to query DNS and maintain SRV records of all the available SIP proxies in the corresponding domain. For each SIP client's request, the LB is responsible to forward the request to the most appropriate SIP proxy, in terms of workload, to serve it. In other words, as described in Figures 4 and 5, SIP clients firstly communicate with the LB entity to find out the best SIP proxy available. If the LB is not responding the SIP client can communicate directly with the DNS to retrieve all the available SRV records corresponding to SIP servers in the domain and select one [9].

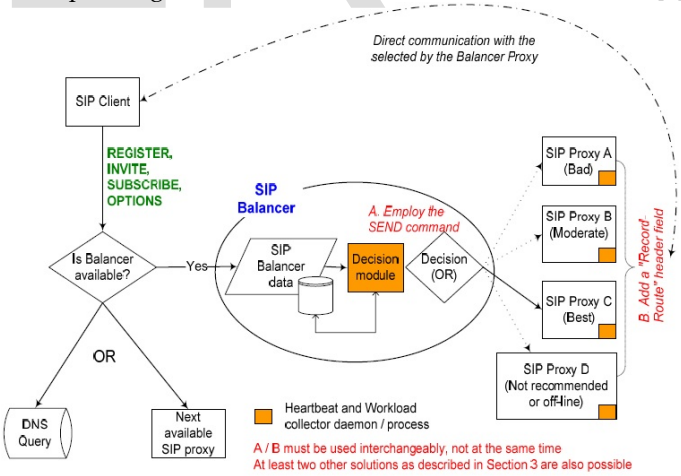


Fig. 5. SIP LB Decision Flowchart

Most SIP clients do not support DNS direct transactions; another solution for them is to communicate directly with another available SIP proxy in the same domain. The IP addresses of the LB and the backup SIP proxies can be pre-configured in the SIP client device. As a result, the IP address of the most appropriate SIP proxy is selected by the LB and while the initial message (e.g. INVITE) goes through the LB, the

subsequent messages for the same session go directly to the selected by the LB SIP proxy[10].

2) Multiprotocol label switching network multipath routing (MPLSMR) Policy:

MPLS- Traffic Engineering (MPLS –TE) is an advanced packet-forwarding technique uses encapsulated fixed length labels to make high speed forwarding decisions [11, 22]. Routers in MPLS network are referred as Label Switching Routers (LSRs). The router by which a packet enters the MPLS is called the ingress LSR, and the one by which it leaves the MPLS is called the egress LSR. Label switched path (LSP) is the routing path that starts with ingress Label switch router and terminates at egress LSR. The ingress nodes add the label for each incoming packet. The labels are detached from the egress border node, whenever the packet leaves the MPLS domain. With MPLS, the processing overhead required for routing at the intermediate nodes could be reduced, thereby improving their packet forwarding performance. In addition MPLS-TE approach [11] allows setting up explicitly routed Traffic Engineering-Label Switched Paths (TE-LSPs) whose paths satisfy a set of traffic engineering constraints, including bandwidth, throughput etc. The example of MPLS network with an LSP from LER1 to LER2 for the flow f1 from server s1 to destination host h1 is shown in figure 6.

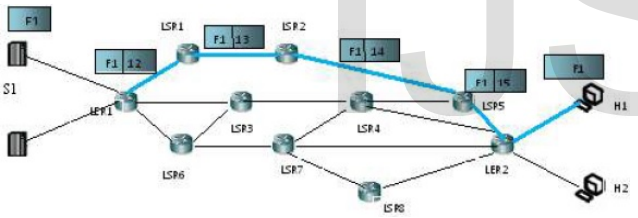


Fig 6. MPLS Network

Load balancing moves the traffic from congested links to other parts of the network. Link or node failures introduce additional overhead to reconfigure routing tables and find an alternative path for packet dispersion. The objective of traffic aware routing and load balancing are to avoid congestion while the traffic is routed from a router to another.

To utilize all the available paths efficiently, this approach first finds an intermediate node k, from which multiple paths satisfy the QoS constraints are discovered. The packets are dispersed into these multiple paths with the required QoS. An example for multipath routing is shown in figure 7. Multipath routing involves flow classifier, path discovery (scheduler), traffic distribution, splitting and Load balancing.

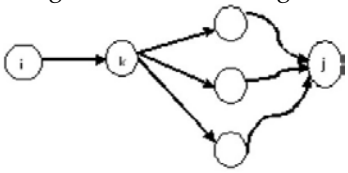


Fig 7. Multipath from intermediate node to destination

The link failure or system failure in IP network causes load unbalanced situation and increases the congestion in the network. During this load unbalanced condition both responsive and unresponsive VoIP flows are rerouted into congestion free constraint based least loaded multiple paths. The non-VoIP data flows are routed using Stream Control Transmission Protocol Multi-homing technique. Flow arrival rate, packet loss rate and delay are measured and taken as the input parameters and compared with the threshold values to identify the VoIP flow. Network load status is calculated by estimating the average buffer occupancy value and multipath routing is triggered when the network load is high to enhance the QoS. We will explore the most interesting teqniques which use MPLS policy [22]:

2.1) Multipath Adaptive Forwarding Equivalence Class (MAFEC) Algorithm:

This algorithm transmit the same label traffic into multiple available LSPs (label switch path). This technique tries to avoid routing oscillation and aims to increase efficient usage of network bandwidth. Fish network model is used to analyze and find multiple paths link disjoint paths. The packets are transmitted on all available shortest LSP to perform link utilization. Efficient Bandwidth Estimation Management is proposed for VoIP Concurrent Multipath Transfer [12]. Grouping-based Multipath Selection algorithm is utilized for selecting the multiple paths for packet dispersion. The bandwidth for each path is computed using Westwood approach. Then the best paths that satisfy the required QoS selected and the packets dispersed into multiple paths using Stream Control Transmission (SCTP) protocol. Packets are dispersed into one shortest path and the remaining is considered as backup paths. The path failure or node failure utilizes the backup paths and bandwidth sharing among these paths increases network resource utilization. A reliable multipath routing for IEEE 802.16 wireless mesh networks was selected [12] to perform interference load aware (ILA) routing. The source node selects multiple paths for the destination node and a value is associated with each path by evaluating QoS constraints. One path is notified as primary path and the alternate path is chosen from the set of multiple paths which has the second minimum value. In case of failure in alternate path, then source node chooses second alternate path that has next minimum value and this process continues. This technique tries to avoid congestion and packet losses.

2.2) Randomized Distance Vector Routing Protocol (RDVRP) Algorithm:

RDVRP is used to distribute the data traffic randomly over all available paths to a destination in the network to perform load balancing. Multiple loop-free next-hops is found for a

destination and updated with the forwarding table (FIB) at each router. Keeping multiple entries will increase the memory needed for FIB. The RDVRP describes the changes in the routing tables [13].

2.3)Fauzia Idrees Approach:

This technique depends on analyzing the currently existing voice applications like Skype, Google Talk, Yahoo voice, MSN voice. The packets from different VoIP services are compared with the non-VoIP packets like E-mail, file downloading, file sharing, instant messaging, games and video. They investigated packet-inter arrival time, average packet size, rate of packet exchange and packet exchange sequences to identify VoIP traffic. From the observations, they conclude that packet size and average number of packets received per second for VoIP applications are noticeable factors when compared to non-VoIP applications. The number of voice packets received per second is between 20 and 40. For other applications it is between 2 and 10. The average packet size is between 100 and 250 bytes whereas other applications packet size is more than 400 bytes. Hence the VoIP traffic detection algorithm first filters the User Datagram Protocol (UDP) packets and depending on whether the packet count lies between 20 and 40 per second and identifies them as VoIP traffic. A multi service differentiation model was applied by defining three types of paths to be traversed by traffic flows. They classify the flow and assign the path, based on the threshold values. A standard path defines the shortest route available between a pair of source and destination nodes and allocated for priority flows. Alternative path generally corresponds to a path which is longer than the standard paths and allocated for non-priority flows. Null path defines a route traversed by non-priority data flows in which packets can be dropped. The resources at each node are monitored by assessing the Differentiated Services (DiffServ) queue lengths. When the queue length reaches deviation threshold it acts as a pre-congestion alarm and all the previously assigned paths are maintained and new flows are assigned to alternate paths. The critical threshold triggers the packet dropping process and yields a new set of null paths which will route demoted non-priority data flow. The standard threshold indicates that a steady traffic load situation is reached and paths are available again to all new incoming traffic flows [13].

2.4) Traffic splitting approach in MPLS networks:

Jose M. F. Craveirinha et al. [14] have proposed a traffic splitting approach in MPLS networks. With this approach, a given node-to-node traffic flow is divided into two disjoint paths based on the load balancing cost and available link bandwidth. Load balancing cost is measured as bandwidth occupied in the paths. Least utilized paths (LSPs) are estimated using path ranking algorithm with minimum number of links in the path and load balancing cost as criteria. If there are alternative optimal paths, dominance test is used to select the appropriate path based on the threshold values.

2.5) Bandwidth engineering (BE) Algorithm:

Bosco A. et al. [15] proposed this approach to improve the MPLS control plane functionalities. BE aims Traffic Engineering (TE) mechanism to work in off-line and on-line mode. The off-line routing is implemented with the global path-provisioning module to compute LSP for the given traffic flow. For high priority traffic flow, the path is estimated in the off line mode and can't be changed. When the load increased, the bandwidth is increased without delay, by setting up a local admission control element. To offer the bandwidth attribute of a high priority flow, it verifies the availability of bandwidth from the other LSPs or even from the low priority flows and allocates to the high priority flows. For low priority flow, the LSP is estimated in off-line mode but it can be changed dynamically when the network load changed. Even the low priority flow is suspended.

2.6) MP-VoIP &OLU Algorithm:

Faritha Banu1, K. G. Shanthi2, P. Lakshmi Priya3, M. Faritha designed an effective technique for load balancing with optimal link utilization (MP-VoIP) to forward the VoIP packets into the specifically selected and guaranteed QoS multiple paths in contrast to the traditional single path approach. This technique mainly deals with the flow classification and implements the load adapter [15].Flow classification algorithm resides in the data plane that classifies Internet traffic flows as VoIP flow and Normal data flow. A label field was assigned for each classified packet. Traffic prioritization and bandwidth management algorithm reside in the management plane that monitors and performs traffic shaping to ensure QoS at each node. Load adapter resides in the control plane that evaluates the network load balancing state to make the packet forwarding decision. The steps involved that work are depicted in the block diagram shown in Fig 8.

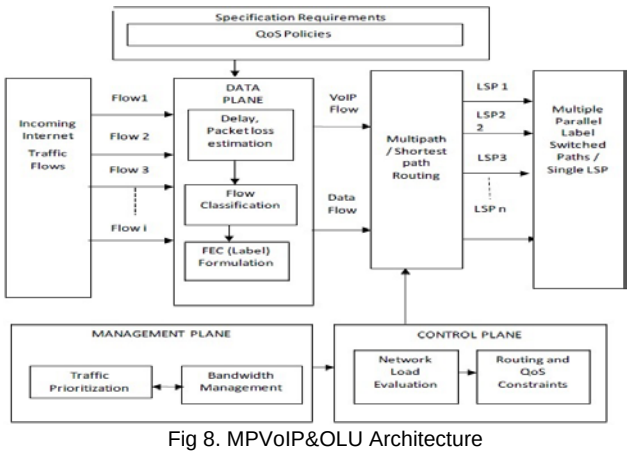
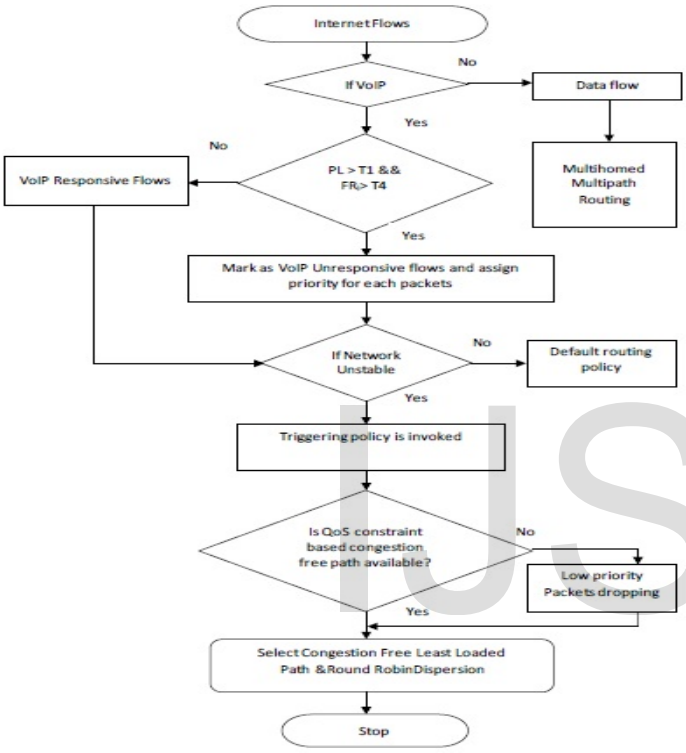


Fig 8. MPVoIP&OLU Architecture

A multipath routing policy is adopted based on network load condition with traffic engineering constraints such that packets traversing the network are experienced with minimum delay and maximum bandwidth. Control plane calculates the number of paths that can be provided for each incoming traffic flow according to its specific QoS requirements and current network topology. Then, the incoming traffic is divided into these paths based on the available bandwidth and delay experienced by each path. MPLS default classifier agent is modified to identify the VoIP and Data flow separately. When the flow enters into the MPLS edge node, MPLS packet



classifier identifies the label field, if the label is not assigned then the control is passed to the modified classifier agent. Flow classification algorithm considers the threshold value for packet loss rate, delay and flow arrival rate. Threshold values are compared with the estimated values to identify the flow, and the packets are added into the corresponding queue for scheduling.

ITU standard G.711 voice codec produces 50 packets per sec with payload size of 160 bytes, packetization period of 20 ms and sample interval of 10 ms. The VoIP packet format is shown in Figure 9.

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 9. VoIP packet format

The packet is encapsulated by appending 18 bytes of Ethernet (link layer) header, 20 bytes of IP header, 8 bytes of User Datagram Protocol (UDP) header and 12 bytes of Real time Transport Protocol (RTP) header with the original voice payload. Hence the VoIP packet size is estimated as 218 bytes. The current Internet voice conversations like skype, Google Talk, Yahoo voice and MSN VoIP have 25, 21, 28, 36 packets per second respectively [15] . The maximum number of packets for G.711 codec is 50 and requires (218 × 50 × 8 = 87200 bits per second) 87.2 Kbps bandwidth. The current VoIP transmits or receives more than 15 packets per second (218 × 15 × 8 = 26100 bits per second). A threshold value of flow arrival rate (T3) is taken as the range from 26.16 kbps to 87.2 Kbps and threshold value of delay T2 is greater than the packetization period (25 ms). The code for the Flow Classification algorithm1 is given as follows.

INPUT: Packet loss rate PLr, Delay Dl, Flow arrival rate FRi, Threshold value T1 = 0.01(PLr), Threshold value T2 = 25 ms, Threshold value 26.16 kbps < T3 > 387.2 Kbps

PROCEDURE:
For a given set of network paths from source to destination P1, P2, P3… Pi € P
For (every node h in current path P)
For (every incoming Internet Flow)
Do

If (packet loss rate > T1 && delay > T2)
If (If (Flow arrival rate > minimum threshold value of T3 && Flow arrival rate < Maximum threshold value of T3)
Step 6.1: Flow marked as “VoIP Flow”
Else
Step 7.1: Flow marked as “Normal Data Flow”
End
OUTPUT:
Marked VoIP flow, Data flow

Fig. 10 Flow chart for unresponsive voice transmission technique

A flow can be described as a sequence of packets exchanged between two nodes. To analyze the flow characteristics, relevant information about the specific flow is stored rather than storing information about each packet. This per flow state information is used to find congestion free optimal path selection. Unresponsive flow identification module operates in the network layer that analyses the packet headers and payloads of each packet to estimate the flow arrival rate. Packet loss rate is estimated by sending a continuous query message at regular time interval of 100 ms along the label switched path P from the ingress router to the egress router. Later, ingress router sends a query message indicating the number of packets transmitted at time t1 along path P [16]. The egress router receives the request query message and sends a reply message with the count of number of packets received along path P at time t2. The number of the lost packet is the difference between received packets and transmitted

packets. Packet loss rate at various time intervals is calculated as the sum of difference between number of test packets received at the egress node and number of test packets transmitted at the ingress node. Packet loss rate is estimated as given in Equation (1).

$$PL_t^p = \sum_{1}^{t} R_{ti+1}^p - T_{t1}^p \quad (1)$$

The threshold values T1, T2 and T3 are introduced in MPVOIP. Threshold value of T4 defines maximum unresponsive flow arrival rate. If the packet loss rate exceeds the threshold T1 (packet loss rate of 1%), flow arrival rate FRi exceeds minimum threshold value T4 (87.2 Kbps < T4 <176 Kbps), then the flow is marked as unresponsive VoIP flow.

Load adapter estimates the average buffer occupancy value (Bavg) to evaluate the network load [17]. The average buffer occupancy value is used to estimate the probability Pli for unresponsive VoIP flow. The packet loss probability depends on the average buffer size (Bs). Let the packet count (Pk), the number of packets queued for the flow for each unresponsive flow is known. The probability constant Pc is estimated as given in the Equation (2). The probability Pli is estimated as given in Equation (3).

$$\text{Probability constant } P_c = \left(B_{avg}\right) * Q / \left(Q - \left(B_{avg}\right)\right) \quad (2)$$

$$\text{Probability } P_{li} = \left| P_c / \left(1 - P_k * P_c\right) \right| \quad (3)$$

Fig 11. Modified label format.

Where Q is the threshold value represents maximum queue size of 75% of the total buffer capacity (Bs). The unresponsive flow is routed through the least loaded congestion free multiple paths to balance the network load. A prior estimation of the bandwidth, delay ensures QoS guaranteed path selection. To achieve load balancing, the incoming traffic is divided into traffic split ratio (αi) along the selected multiple paths on a per packet basis by adding an identifier. Traffic split identifier can be inserted in the TTL (Time-TO-Live of the packet till ends and dropped) field of the MPLS header as proposed by Avallone et al. [17]. Figure 11 shows the utilization of TTL filed to insert the traffic split identifier. Link failure or system failure takes a considerably long time to update its routing table. This leads to increase in congestion and packet losses. Let the core router monitors the congestion and alerts the nodes periodically. During the occurrences of congestion, this priority marking and policy dropping shapes the flow by dropping the packets from the low priority unresponsive flow to reduce congestion. Since the priority is assigned for all the unresponsive VoIP flow at the ingress router, it reduces the complexity of traffic shaping in core router.

This technique identifies the normal data flow and VoIP flow separately from the incoming traffic flows by measuring Flow arrival rate, packet loss rate and delay and queues up for routing with MPLS label identifier. Quality of Service for VoIP flows are improved by routing the VoIP flow through multiple paths which satisfy the given input constraints. The VoIP flow whose arrival rate and packet loss rate are higher than the specific threshold value is classified as unresponsive and low priority is set up for those packets.

3) Information based-Call Allocation policy

These strategies are grouped by the type and amount of information used for allocation:

(a) knowledge-free (KF): with no information about applications and resources.

(b) utilization-aware (UA): with CPU utilization information.

(c) time-aware (TA): VM rental time information (beginning and completion).

(d) load-aware (LA): VM load information in two time frames.

Table 2 summarizes the call allocation strategies that request new VM when arriving call exceeds the threshold of utilization in all VMs. Table 3 shows the call allocation strategies with load estimation. Each time interval, the strategy makes an estimation of utilization for next time interval and if it overpasses the threshold current VMs, it requests new VM [18].

Algorithm 2 describes the BFit_xx strategy, where Voice Nodes

Label(20bits)			Class of Service (CoS)3 bit	S 1bit	Traffic Split id8 bits
5 (bits) Sequence Number	3 (bits) Priority	12 (bits) Unused			

(VNs) are separated in two lists: Admissible Voice Node List (AVNL) and no AVNL list (nAVNL). AVNL list contains the VNs that their finish time is not less than in xx minutes (time-aware). Both lists are sorted in not decreasing order of their utilization. When the VNs on AVNL cannot process the arriving calls due to exceeding the utilization threshold, the strategy searches on nAVNL. If a VN is available to process calls without QoS degradation then the call is placed to it, the VN is moved to AVNL and it schedules one hour more of rental time.

TABLE 2
CALL ALLOCATION STRATEGIES

		Description
KF	Rand	Allocates job *j* to VM randomly using a uniform distribution.
	RR	Allocates job *j* to VM using a Round Robin algorithm.
UA	FFit	Allocates job *j* to the first VM capable to execute it.
	BFit	Allocates job *j* to VM with smallest utilization left.
	WFit	Allocates job *j* to VM with largest utilization left.
TA	MaxFTFit	Allocates job *j* to VM with farthest finish time.
	MidFTFit	Allocates job *j* to VM with shortest time to the half of its rental time.
	MinFTFit	Allocates job *j* to VM with closest finish time.
	Rand_05 Rand_10 Rand_15 RR_05 RR_10 RR_15	Allocates job *j* to VM that finishes not less than in 5, 10, 15 minutes using the Rand, and RR strategies.
UA + TA	BFit_05 BFit_10 BFit_15 FFit_05 FFit_10 FFit_15 WFit_05 WFit_10 WFit_15	Allocates job *j* to VM that finishes not less than in 5, 10, and 15 minutes using the Bfit, FFit, and WFit strategies.

TABLE 3
CALL ALLOCATION STRATEGIES WITH PREDICTION

When the call cannot be assigned to VNs without QoS degradation, the strategy attempts to allocate the call on the nAVNL without QoS guarantee. If not, the call is placed into the call queue waiting for a new VM. The main goal of the algorithm is to use running VNs, even if they in nAVNL list, instead to start new VNs instances. The startup time reduces the QoS, so algorithm looks first on nAVNL list. In the worst case, algorithms start a new VN.

Algorithm 2. Best Fit TA (BFit_xx)

Input: Voice node list (VNlist), timeAware (TA) and call.
Output: Allocation of call in one voice node.
vnIndex ← -1
Create AVNL and nAVNL lists with TA time.
Sort AVNL by utilization on decreasing order.
Sort nAVNL by utilization on decreasing order.
vnIndex ← Best_Fit(AVNL, 0.7, call)
if vnIndex < 0 then
vnIndex ←Best_Fit(nAVNL, 0.7, call)
if vnIndex < 0 then
vnIndex ←Best_Fit(nAVNL, 1.0, call)
if vnIndex < 0 then
vnIndex ←Best_Fit(AVNL, 1.0, call)
if vnIndex < 0 then
Send call to call_queue
Start a new node voice
else
Insert call into VN with index vnIndex
Endif

4) VM-Aware Adaptive Rate of Change (VMA-AdRoC) Policy:

Campos and Scherson designed for making job distribution decisions at runtime by using load balancer BAL, locally and asynchronously. RoC-LB [19] is a dynamic distributed load balancing algorithm, it achieves the goal of minimizing processor idling times without incurring into unacceptably high load overheads. Resources calculate the change in their load between two sample intervals. The Sampling Interval (Si) is an adaptive parameter, and its length may vary. Each resource asynchronous calculates the Difference in Load (Dl), and use it as estimation on load for the next Si, this estimation allows to allocate and reallocate jobs more efficiently. A finer sampling allows to improve the balance the system, but it increases the overhead. The concept of Dl was used as a mechanism to predict requests for new VMs, which can be provided after StUp time. Dl (utilization amount of arriving calls) are used to estimate the number of VMs after Si. It permits to initialize VMs before the arriving calls degrade the QoS.

Let ui(t) be the utilization of SNCi (Super Node Clusters) at time t, and Ki(t) the number of VMs running, then the rate of load change during the sample interval Si=[t− Si,t] is defined

		Description
LA	Rand_stUp Rand_s10 Rand_s20 Rand_s30 RR_stUp RR_s10 RR_s20 RR_s30	Allocates job *j* to VM using the Rand, and RR strategies. They use intervals of 10, 20, 30 and stUp seconds to estimate future load
UA +LA	BFit_stUp BFit_s10 BFit_s20 BFit_s30 FFit_stUp FFit_s10 FFit_s20 FFit_s30 WFit_stUp WFit_s10 WFit_s20 WFit_s30	Allocates job *j* to VM using BFit, FFit, and WFit strategies. They use intervals of 10, 20, 30 and stUp seconds to estimate future load

by: Δi(t) = (ui(t) − ui(t−Si))/Ki(t).

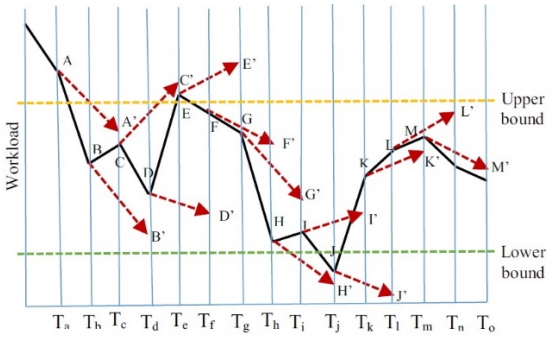


Fig. 12. VMA-AdRoC Prediction scenario

Fig. 12 shows Dl changes scenario. Solid line represents real workload and dashed lines are estimated workloads. If the load is larger than Upper bound (Ub) then the broker request for a new VM. If the predicted load is less than Lower bound (Lb) the broker can reduce the amount of running VMs. Ub and Lb are adjustable parameters than depend on the number of running VMs and the utilization threshold. At time Tc, the broker immediately initiates a new VM based on predicted future load. Estimation J' on Tj is under Lb but broker cannot initiates a reduction process because the request at Th is not finished yet. New request can be generated only after Th [20].

To define WHERE a load is requested from or send to, each BAL keeps two lists. The sink list records Bals that previously needed jobs, and source list enrolls Bals that previously offered jobs. BAL that initiates a request is considered to be a sink. A sink selects a BAL from its source list for a load request, and sends a requesting message. The source can accept the request or broadcast the request to other Bals from its own source list. BAL does not send several load requests at the same time. It has to wait an answer for the first request until it sends another message. The result of this message is the load coming from other BAL or the request comes back as unfulfilled.

4 Our Model: Scalable Voice Over Internet (S-VoIP) Model

Installing VoIP System over Cloud:

3CX Phone System is a software-based IP Private Branch Exchange (PBX) that replaces a traditional PBX and delivers employees the ability to make, receive and transfer calls [6]. The IP PBX supports all traditional PBX features. An IP PBX is also referred to as a VOIP Phone System, IP PABX or SIP server. Calls are sent as data packets over the computer data network instead of via the traditional phone network. Phones share the network with computers and separate phone wiring can therefore be eliminated. With the use of a VOIP gateway, we can connect existing phone lines to the IP PBX and make and receive phone calls via a regular PSTN line. 3CX phone system uses standard SIP software or hardware phones, and provides internal call switching, as well as outbound or inbound calling via the standard phone network or via a VOIP service.

One of the main benefits of 3CX is the scalability which means ease to outgrow and expand. Adding more phone lines or extensions often requires expensive hardware upgrades. In some cases we need an entirely new phone system. Not so with a VOIP phone system: a standard computer can easily handle a large number of phone lines and extensions – just add more phones to our network to expand.

3CX has Others benefits such as much easier to install, configure then a proprietary phone, easier to manage , call cost reduction, no need for separate phone wiring, use computer network, no vendor lock-in, Better customer service & productivity, Web based Switchboard makes phones easier to use, Better control via better reporting..etc.

Our model consists of one (or more SIP standard based phones), an IP PBX server, and optionally a VOIP Gateway. The IP PBX server is similar to a proxy server: SIP clients, being either soft phones or hardware based phones, register with the IP PBX server, and when they wish to make a call they ask the IP PBX to establish the connection. The IP PBX has a directory of all phones/users and their corresponding SIP address and thus is able to connect an internal call or route an external call via either a VOIP gateway or a VOIP service provider. Our model use SIP protocols. Virtual Machines (VMs) execute 3CX, and provide calls, voice mails, video/audio conferences, interactive phone menus, call distribution, etc. Additionally, users can transfer images and texts, and they can create new functionalities, opening up a complete new experience in telephonic communication. It is framework, under free license, for building multi-protocol, real-time communication solutions providing a powerful control over call activity. Each VoIP node has 3CX running process with unique IP address that is used by end users to connect. See figures 13:17.

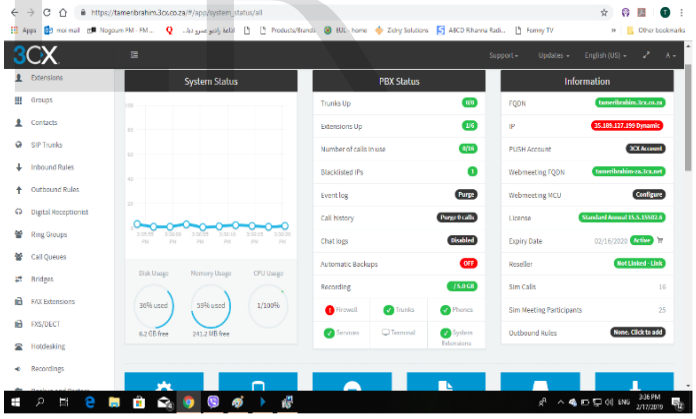
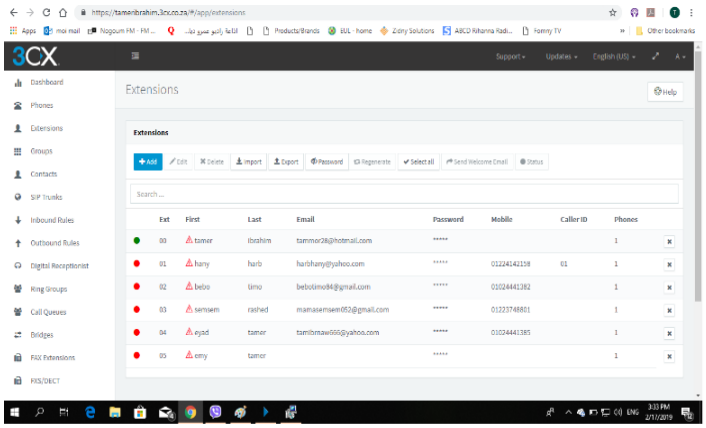


Fig. 13 S-VoIP Home page

Fig.14 S-VoIP Extensions
Fig.15 S-VoIP calls



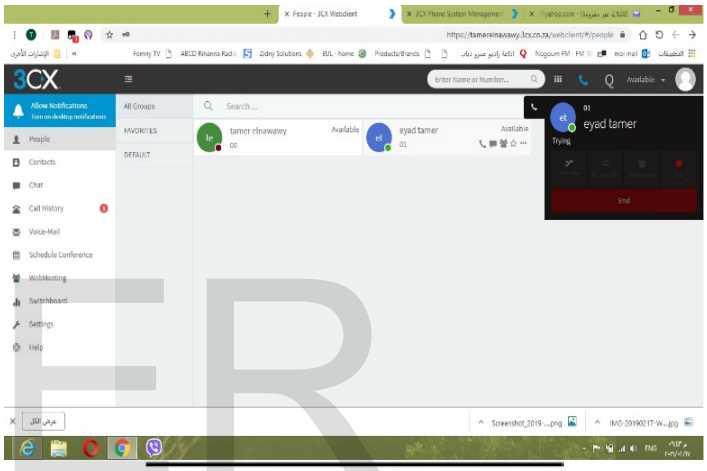


Fig. 16 Client 1 Hard Phone S-VoIP

Fig. 17 Available Clients on Hard Phone S-VoIP

Scalability of our model:

VoIP Calls have different impact on the processor utilization [21] depending on the operations performed by 3cx. Table 4 shows processor utilization for call without transcoding presented by Montoro et al. 2009, see Table 4.

TABLE 4
UTILIZATION FOR CALLS WITHOUT TRANSCODING

Protocol	Codec	10 Calls	1 Call
SIP/RTP	G.711	2.36%	0.236%
SIP/RTP	G.726	2.13%	0.213%
SIP/RTP	GSM	2.58%	0.258%
SIP/RTP	LPC10	1.92%	0.192%

We addressed S-VoIP in distributed cloud environment with high heterogeneity of the resources with different number of servers, execution speed, energy efficiency, amount of memory, bandwidth, etc. We considered our S-VoIP cloud infrastructure consists of m heterogeneous SNCs (Super Node Clusters): SNC1, SNC2…, SNCm with relative speeds S1, S2….Sm. Each SNCi for all i=1…m, consists of mi SNs. Each for all k=1…mi, runs Ki (t) VM at time t. We assumed that VMs of one SNC are identical and have the same processing capacity. We denoted the number of billing hours in SNCi by:

$\bar{m}_i = \int_{t=0}^{C_{max}} k_i(t) \cdot m_i \, dt$ and run in all SNC $\bar{m} = \sum_{i=1}^{m} \bar{m}_i$ by

The VM is described by a tuple {vmui (t)}, where vmui (t) is the utilization (load) of the VM at time t. VM hosts one or multiple 3CX running processes that run and handle calls. The SNC contains a set of routers and switches that transport traffic between the SNs and to the outside world. They are characterized by the amount of traffic flowing through it (Mbps). A switch connects a redistribution point or

computational nodes. The connections of the processors are static but their utilization is changed. The SNC interconnection network architecture is local. The interconnection between SNCs is provided through public Internet.

We considered n independent calls or jobs J1, J2, ...Jn that must be scheduled on set of SNCs. The job Jj described by a tuple {rj, pj, uj} that consists of: its release date rj≥0 duration pj (lifespan), and contribution to the processor utilization uj. The release time of a job is not available before the job is submitted, and its duration (time) is unknown until the job has completed. The utilization is a constant for a given job that depends on the used codec and normalized for the slowest machine [22].

In our S-VoIP, CPU utilization is a key performance metric for VoIP quality of service measurement. It can be used to track QoS regressions, when it increases above the certain threshold, or improvement, when it is below, and is a useful for VoIP QoS problem studying. We considered that the node size is equals to 1 that corresponds to 100% of VM utilization. We used an adaptation of five known strategies First-Fit (FFit), Best-Fit (BFit), Worst-Fit (WFit), Round Robin, and Random in order to allocate calls to VMs. We sorted nodes in decreasing order by their utilization in the BFit strategy. After allocation phase, we have a set of VMs that host one or multiple 3CX running processes and calls. Each VM is characterized by a relative execution speed, and current utilization (load).

We used NS2: a framework for modeling and simulation of cloud computing infrastructures. It is a standard trace based simulator that is used to study cloud resource management problems. We extended the NS2 by introducing the support of dynamic arrival of the jobs (calls), updating the system parameter before scheduling decisions (utilization of the resources), and implementing the broker policies for call allocation. Parameters are directly taken from traces of real VoIP service.

The proposed workload is a set of registered phone calls that have been handled by the system. It is recorded in the Call–Detail–Record (CDR) database with the following information: Index of the call, ID of the user who makes the call, IP of the phone where the call is placed from, IP of the local phone, Destination of the call, Destination country code, Destination country name, Telecommunications service provider, Beginning of the call (timestamp), Duration of the call (in seconds), Duration of a paid call, Cost per minute, etc. Supported call-statistics include: Incoming/outgoing call attempts, whether successful or not; Calls rejected or failed; Number of calls whose connected time is less than the configured minimum call duration (MCD); Number of calls losing more than the configured number of packets; Number of calls encountering more than the configured amount of latency, jitter; calls disconnected; etc.

For the analysis, we used 30 workloads; each includes phone calls made during one day. First, we analyzed the mono-objective problem, where a utilization threshold is used to guaranty the QoS. Then, we realized a bi-objective analysis, where no threshold is used (100% of utilization is allowed) to study the relation between total cost (the number of hours running VMs) and the utilization of the VMs. We evaluated the performance of the five strategies with a 70% utilization threshold of VMs to ensure the QoS. Figure 18 shows the number of billing hours during a month. The strategies have similar behavior when workload is low (during weekends). During the week, strategies produce significally different cost. Bfit and FFit use about 55 billing hours, while Rand, RR, and WFit use about 75 hours.

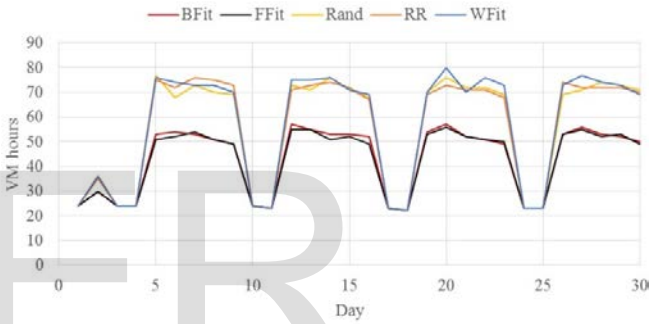


Fig. 18 The number of billing hours during 30 days

Figure 19 shows an example of the number of billing hours during a day. The workload is low during the first hours of the day, all strategies have the same behavior and they use just one VM to process the calls. The maximum number of VMs running during peak hours is 5 VMs. Figure 20 shows the average number of billing hours during 30 days. FFit and BFit are the best strategies. They used 42.5, 43.1 billing hours on average.

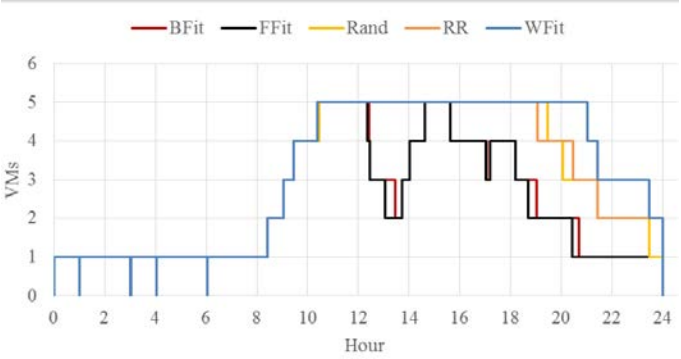


Fig. 19 Example of the number of billing hours during a day.

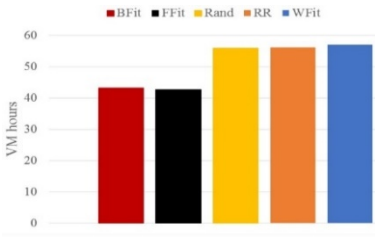


Fig. 20 Average billing hours per day

The monthly difference between these strategies Rand, RR, and WFit is about 14 hours per day. Rand and RR need 56.9 and 57.1 billing hours. The worst strategy is WFit with 57.9 billing hours per day on average.

5 CONCLUSIONS

In this paper, different models were formulated and discussed to address job allocation problem associated with VoIP calls in cloud computing. Our model was designed using our own software 3CX and NS2 network simulator to measure and identify the optimum approach of balancing the load of VoIP traffic. Provider's cost and quality of service models were defined in this paper, and a customized approach/strategy for VoIP super node clusters was proposed. Environment's uncertainty, QoS and cost optimization were considered in our proposed model and strategies. Allocation decisions are taken based on the actual cloud and VM characteristics at the moment of allocation such as number of available virtual machines, their utilization, etc. Due to the dynamicity of these over time, our allocation strategy adapts to these changes. Our approach is reliable and scalable. It copes with different workloads, cloud properties, and cloud uncertainties such as elasticity, performance changing, virtualization, loosely coupling application to the infrastructure, and other variables (e.g., effective processor speed, number of available virtual machines, and actual bandwidth, etc.). The proposed approach/algorithm can be used for a VoIP cloud environment. However, further study is required to assess its actual efficiency and effectiveness in each domain. This is to be covered in a future work. In addition, dynamic consolidation and load balancing is another important issue to be addressed.

REFERENCES

[1] L. Madsen, J. V. Meggelen, and R. Bryant. Asterisk: The definitive guide. O'Reilly Media, Inc., 2011.

[2] H. P. Singh, S. Singh, J. Singh, and S. A. Khan. VoIP: State of art for global connectivity—A critical review. Journal of Network and Computer Applications, 37, 365-379, 2014.

[3] J. M. Cortés-Mendoza, A. Tchernykh, A. M. Simionovici, P. Bouvry, S. Nesmachnow, B. Dorronsoro, and L. Didelot. VoIP service model for multi-objective scheduling in cloud infrastructure. International Journal of Metaheuristics, 4(2), 185-203, 2015.

[4] M. Mao and M. Humphrey. A performance study on the vm startup time in the cloud. In Cloud Computing (CLOUD), IEEE 5th International Conference on, 423-430, 2012.

[5] K. Razavi, L. Razorea, and T. Kielmann. Reducing vm startup time and storage costs by vm image content consolidation. In Euro-Par 2013: Parallel Processing Workshops, 75-84, Springer Berlin Heidelberg, 2013.

[6] 3CX Phone System and ATOM N270 Processor Benchmarking. http://www.3cx.com/blog/voip-howto/atom-processor-n270- benchmarking, accessed September 20, 2016.

[7] A. Montazerolghaem, S. Shekofteh, M. Yaghmaee, and M. Naghibzadeh. A load scheduler for SIP proxy servers: design, implementation and evaluation of a history weighted window approach. Int. J. Commun. Syst, 2015.

[8] A. Montazerolghaem, M. Hossein, A. Leon-Garcia, M. Naghibzadeh,and F. Tashtarian. A Load-Balanced Call Admission Controller for IMS Cloud Computing. IEEE Transactions on Network and Service Management, 2016.

[9] W. Song, Z. Xiao, Q. Chen, and H. Luo. Adaptive resource provisioning for the cloud using online bin packing. Computers, IEEE Transactions on. 63(11): 2647-2660, 2014.[10] A.Wolke, B. Tsend-Ayush, C. Pfeiffer, and M. Bichler. More than bin packing: Dynamic resource allocation strategies in cloud data centers. Information Systems, 52, 83-95, 2015.

[11] Y. Li, X. Tang, and W. Cai. Dynamic bin packing for on-demand cloud resource allocation. Parallel and Distributed Systems, IEEE Transactions on. 27(1):157-170, 2016.

[12] http://blog.cloud66.com/ready-steady-go-the-speed-of-vm-creationand- ssh-key-access-on-aws-digitalocean-linode-vexxhost-googlecloud- rackspace-and-microsoft-azure/, accessed September 20, 2016.

[13] CloudSim: A framework for modeling and simulation of Cloud Computing infrastructures and services. http://www.cloudbus.org/cloudsim/, accessed Sept. 20, 2016. [14] A.M. Simionovici, A. A. Tantar, P. Bouvry, A. Tchernykh, J. M. Cortés-Mendoza, L. Didelot. VoIP traffic modelling using Gaussian mixture models, Gaussian processes and interactive particle algorithms. In 2015 IEEE Globecom Workshops, 1-6, 2015.

[15] https://www.mixvoip.com/, accessed September 20, 2016.

[16] Andrei Tchernykh, Luz Lozano, Uwe Schwiegelshohn, Pascal Bouvry, Johnatan E. Pecero, Sergio Nesmachnow, Alexander Yu. Drozdov. Online Bi-Objective Scheduling for IaaS Clouds with Ensuring Quality of Service. Journal of Grid Computing. Springer-Verlag, vol. 14, Issue 1, 5–22, 2016.

[17] J. M. Cortés-Mendoza, A. Tchernykh, A. Yu. Drozdov, and L. Didelot. Robust cloud VoIP scheduling under VMs startup time delay uncertainty. 9th International Conference on Utility and Cloud Computing, 234-239, 2016.

[18] http://www.mobicents.org , accessed November 10, 2016.

[19] https://aws.amazon.com/es/solutions/case-studies/?nc2= h_ql_ny_livestream_blu, accessed November 10, 2016. [24] https://cloud.google.com/customers, accessed November 10, 2016.

[20] A. Tchernykh, U. Schwiegelsohn, E.-g. Talbi, M. Babenko. Towards Understanding Uncertainty in Cloud Computing with risks of Confidentiality, Integrity, and Availability. Journal of Computational Science. Elsevier, 2016.

[21] J. M. Cortés-Mendoza, A. Tchernykh, F. A. Armenta-Cano, P. Bouvry, A. Yu. Drozdov, and L. Didelot. Biobjective VoIP Service Management in Cloud Infrastructure. Scientific Programming, vol. 2016, Article ID 5706790, 2016.

[22]Naoum, R.S. and Maswady, M. (2012) Performance Evaluation for VOIP over IP and MPLS. World of Computer Science and Information Technology Journal, 2, 110-11