Optimizing VoIP Call in Diverse Network Scenarios Using State-Space Search Technique

Tamal Chakraborty[1], Atri Mukhopadhyay[1], Suman Bhunia[2], Iti Saha Misra[2], and Salil Kumar Sanyal[2]

[1] School of Mobile Computing and Communication,
Jadavpur University Salt Lake Campus, Kolkata-700098, India
[2] Dept. of Electronics & Telecommunication Engineering
Jadavpur University, Kolkata-700032, India
{tamalchakraborty29,atri.mukherji11,sumanbhunia}@gmail.com,
iti@etce.jdvu.ac.in, s_sanyal@ieee.org

Abstract. A VoIP based call has stringent QoS requirements with respect to delay, jitter, loss, MOS and R-Factor. Various QoS mechanisms are being implemented to satisfy these requirements. These mechanisms must be adaptive under diverse network scenarios. Moreover such mechanisms must be implemented in proper sequence, otherwise they may conflict with each other. The objective of this paper is to address the problem of adaptive QoS maintenance and sequential execution of available QoS implementation mechanisms with respect to VoIP under varying network conditions. In this paper, we generalize this problem as a state-space problem and thereby solve it. Firstly, we map the problem of QoS optimization into state-space domain and then apply incremental heuristic search. We implement it under various network and user scenarios in a VoIP test-bed to optimize the performance. Finally, we discuss the advantages and uniqueness of our approach.

Keywords: VoIP, QoS, State-space, Heuristic, Incremental Search.

1 Introduction

Voice over Internet Protocol (VoIP) [1] has witnessed rapid growth in recent years owing to ease of network maintenance and savings in operational costs. As it is being widely deployed in office and public networks, maintaining the Quality of Service (QoS) of an ongoing call has assumed utmost importance. Network parameters such as bandwidth, error rate, loss rate, latency, etc. varies with time. With increasing number of users, the issues related to admission control, fairness, scalability, etc also need to be properly addressed. So the QoS optimization techniques must be adaptive.

However, abrupt implementation of these techniques without maintaining proper sequence often results in degraded performance. For example, Random Early Detection (RED) buffer is not advantageous without end-to-end congestion control mechanism [2]. Further, it is observed that often such abrupt implementations of optimization techniques conflict with each other. For example, RED implementation for small buffer size is not better than static queue with tail-drop mechanism [2]. However, buffer size must be kept small to reduce delay in real-time traffic during

D.C. Wyld et al. (Eds.): ACITY 2011, CCIS 198, pp. 231–242, 2011.

congestion. Thus they conflict each other. So the decision to apply appropriate optimization technique is crucial.

This paper aims to generalize the problem of QoS implementation amid diverse scenarios by mapping it as state-space problem. The objective is to maintain adaptive QoS in multiple call scenarios and under diverse network conditions by applying available QoS optimization techniques in proper sequence. Focus is also on prioritizing emergency calls with QoS guarantees.

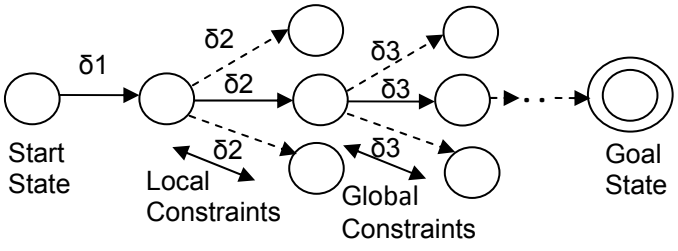
A state space search is the method of finding Goal state(s) from start state through certain intermediate states [3]. In heuristics based search, each state is given a heuristic and traversing is done following a heuristic function. Incremental search further reuses information from previous searches to speed up current search [4]. Incremental heuristic search combines features of both. So it is selected as this aims to fulfill the aforementioned objective.

2 Proposed Mechanism

The aim is to map the problem of optimizing VoIP over various network links into a state-space domain where the next state from a set of intermediate states is selected based on incremental heuristic search obeying certain constraints. This is defined as a tuple [N, A, S, G]. The state-space scenario for each call is shown in Fig. 1.

- 'S' contains the start state which is defined as the call initiation state with respect to time and having heuristics namely delay, loss and Mean Opinion Score (MOS).
- 'G' contains the call termination state as the Goal state with respect to time along with its related heuristics as stated above.
- 'N' contains all the intermediate states within. An intermediate state is taken as any part of an ongoing call with respect to time along with its related heuristics. Heuristics can be categorized as *Excellent*, *Good*, *Average* and *Poor* based on user satisfaction level. Any intermediate state is derived by variation in the network parameters, significant change in heuristic values and by application of QoS optimization techniques.
- 'A' is a set of arcs from one state to another and is effected by transition functions namely $\delta 1$, $\delta 2$ and $\delta 3$. $\delta 1$ is network triggered and can occur due to changes in network. $\delta 2$ is performed by the user in response to $\delta 1$ and involves applying QoS optimization techniques. $\delta 3$ is again user triggered and maintains QoS in a multiple call scenario.

Every heuristic must obey certain local constraints for each call. Delay and loss must be within 180 ms and 5% respectively. MOS must be at least 2. In multiple call scenario, global constraints are taken as mean of local constraints. Now the proposed algorithm is discussed. It consists of two phases, namely ***analysis*** and ***implementation***. Each call at a particular instant of time is taken as state '*s*' and is associated with two metrics namely $g = \{delay, loss\}_{avg}$ and $h = \{delay_{est}, loss_{est}\}_{avg}$. '*g*' calculates the average of delay and loss for already generated states, as measured by network monitoring tool. '*h*' estimates delay ($delay_{est}$) and loss ($loss_{est}$) for the state to be generated following implementation of QoS mechanism.



Index

State-Space	Significance In VoIP
Start State	Call Initiation
Goal State	Call Termination
Local Constraints	Constraints for each call
Global Constraints	Constraints in multiple call scenario
δ1	Change in Network Or Heuristics
δ2	Optimization Technique in single call
δ3	Optimization Technique in multiple call scenario
⟶	Best ranked action
- - - -▶	Other actions

Fig. 1. State Space Diagram for the proposed approach

2.1 Analysis Phase

This phase analyzes all possible conditions of network with respect to delay and loss. There can be 4 scenarios that include delay and loss within tolerable limits, worsening of either delay or loss and finally degradation of both. For each scenario, order of implementation of available QoS mechanisms is selected based on its expected performance. Each such mechanism is denoted by action '*a'*. Mathematically, it is denoted by (1) [5].

$$f = \arg\min_{a \in A(s)} h(succ(s,a)) \tag{1}$$

Successor '*succ'* is next state generated due to application of action '*a'* on state '*s'*. *A(s)* denotes set of available actions to optimize state '*s'*. The best ranked action '*a'* is such that by implementing it, '*h'* becomes minimum as denoted by *argmin* function. There are two other functions namely, *one-of (f)* that describes selection of an action from set of suitable actions and *next (f)* that describes selection of next ranked action from set of suitable actions.

2.2 Implementation Phase

As the call starts, initial state is generated with heuristics. With variation in network parameters or significant change in heuristics, new intermediate states are created. The transition function is termed as *δ1*. Each state is monitored to check whether local constraints are satisfied. Each constraint has a '*threshold'*. If the local constraints are violated, *δ2* is applied to bring the heuristics within *threshold*. This implies that the best ranked mechanism is applied as per analyzed results. New states are monitored.

If local constraints are still not satisfied, next ranked action is implemented and so on.

In multiple call scenarios, global constraints must also be satisfied. Calls with low QoS metrics are classified as '*degraded*' calls and the rest as '*accepted*' calls. Existing QoS implementations for *accepted* calls are stopped temporarily and are redirected to the *degraded* calls. As global constraints are satisfied, new states are generated and corresponding transition functions are termed as *δ3*. All these states are monitored again and QoS mechanisms are implemented to satisfy local constraints. High priority calls are given weights and global constraints are calculated as weighted mean of local constraints. This approach is represented in Fig. 8. The pseudo-code is given under.

1. $s := s_{start}$. //Call initiation state with heuristics.
2. Calculate g_s. //Delay, loss measured for current state
3. IF $s \in G$ THEN GOTO step 18.//Goal state is reached.
4. IF $g_s > threshold$ THEN GOTO step 5 ELSE GOTO step 2.
5. $s := s_{\delta 1}$. Calculate g_s. //New state is generated.
6. $a := one\text{-}of(argmin_{a \in A(s)} h(succ(s,a))$.//Select best action.
7. Execute action a. //Action 'a' is implemented.
8. $s := s_{\delta 2}$. //New state is generated after action 'a'.
9. Calculate g_s. //Delay, loss measured for current state
10. IF $g_s < threshold$ THEN GOTO step 13 ELSE GOTO step 11. /*Local constraints must be satisfied.*/
11. $a := next(argmin_{a \in A(s)} h(succ(s,a))$.//Select next action.
12. Execute action a. GOTO step 8.
13. IF no. of calls>1 THEN GOTO step 14 ELSE GOTO step 3.
14. Classify ongoing calls as *accepted* calls whose $g_s < threshold$. Rest of the calls is *degraded* calls.
15. Stop action $a \in A(s)$ for *accepted* calls.
16. Execute action $a := argmin_{a \in A(s)} h(succ(s,a))$ in degraded calls.
17. $s := s_{\delta 3}$. GOTO step 9.//New state is generated after action 'a'
18. Calculate g_s for $s \in G$.//Goal heuristics calculated

3 Implementation of the Algorithm

3.1 Description of the Test-Bed

Our experimental test-bed, as shown in Fig. 2 consists of fixed and mobile nodes for VoIP communication, wireless access points, a switch and a Session Initiation Protocol (SIP) server. X-Lite [6] is used as the softphone with support for various audio codecs and Group QoS (GQoS). The Brekeke SIP server [7] is SIP Proxy Server and Registrar. ManageEngine VQManager [8] is used to analyze QoS metrics of ongoing call. Further, User Datagram Protocol (UDP) is used with Real-time Transport Protocol (RTP) on top of it.

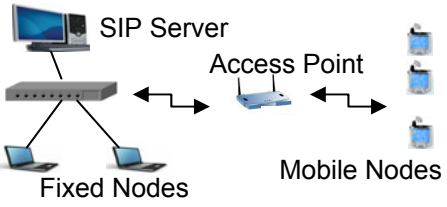


Fig. 2. Experimental Test-Bed

3.2 Test-Bed Analysis Phase

Initially the **effect of buffer size** is studied. Four scenarios are created using Network Emulator for Windows Toolkit (NEWT) [9] as shown in Table 1.

Table 1. Different Network Scenarios

Parameters	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Network Latency in ms	100	100	100	100
Network Loss in %	Nil	Nil	30	30
Buffer Size in packets	Maximum	20	Maximum	20

Table 2. Delay and Loss in each scenario

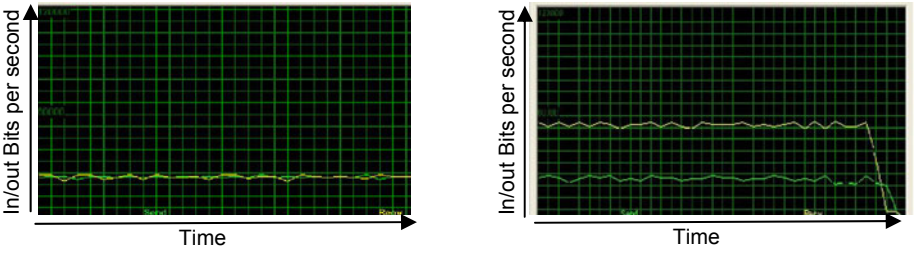
Scenarios	Max. Delay	Avg. Delay	Max. Loss	Avg. Loss
1	156 ms	112 ms	0 %	0 %
2	39 ms	11 ms	44 %	5 %
3	94 ms	6 ms	61 %	22 %
4	6 ms	1 ms	49 %	21 %

As seen from Table 2, increase in loss rate in network results in degraded performance in terms of packet loss in scenario 4. As buffer size is increased, end-to-end delay increases and retransmissions take place after certain timeout, resulting in further loss as in scenario 3. Even in absence of loss rate, increasing buffer size increases end-to-end delay while decreasing it increases loss as seen in scenarios 1 and 2 respectively. So selection of proper buffer size is important.

It is also observed that if **in/out bit rate in an endpoint** (caller/callee) varies significantly with time, call quality drops and terminates at last. BroadVoice 32 (BV32) [10] is used as the codec and in and out buffer size of an end-point is varied to get the results as shown in Table 3. Fig. 3(b) shows the call termination as the in/out bit rate varies in contrast to Fig. 3(a) where the call continues. Thus it can be inferred that similar buffer sizes and hence comparable bitrates must be maintained for a call to continue successfully.

Table 3. Readings for the variable in/out bit rate in an endpoint

Local-> Remote Buffer size (packets)	Remote-> Local Buffer size (packets)	Sending Rate (kbps)	Receiving Rate (kbps)	Call Duration (sec)
20	20	26	26	1137
90	20	26	54	168



(a) (b)

Fig. 3. Effect of (a) constant in/out bit rate (b) variable in/out bit rate on ongoing call

Further, **effect of active queue management** is studied. Active queues drop packets before queue is full based on certain probabilities and threshold parameters to maintain bursts in flows and fairness among users. Here Random Early Detection (RED) [11] queue is implemented by keeping maximum threshold at 100 and minimum threshold at 50. We create two congested media having 1kbps and 10 kbps Constant Bit Rate (CBR) background traffic. As observed from Table 4, in moderately congested medium, delay and loss are within tolerable limits. Thus it is advantageous than having fixed size buffer. However, increase in congestion increases loss when RED is implemented. So selection of active queue management policy is of utmost importance towards maintaining quality of call.

Table 4. Different Execution Scenarios of RED implementation

Parameters	Background traffic 1 kbps	Background traffic 10 kbps
Average Delay in ms	66	70
Average Loss in %	6	14

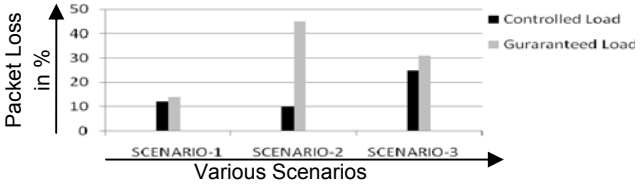


Fig. 4. Effect of Controlled Load and Guaranteed Load on loss in various scenarios

Finally we implement **IntServ model** to optimize VoIP performance. IntServ model proposes 2 service classes [12] namely,

1 Controlled load service [13] for reliable and enhanced best-effort service,
2 Guaranteed load service [14] for applications requiring a fixed delay bound.

Both are implemented in the test-bed for the scenarios as mentioned in Table 1. Experiment results conclude that controlled load service gives better performance in terms of packet loss than guaranteed service in scenarios 1, 2 and 3 as seen in Fig. 4. In scenario 4 which is the most congested scenario, call terminates in 58 seconds and 111 seconds under guaranteed service and controlled load service respectively. So it is concluded that controlled load service is more suited during congestion than guaranteed service.

3.2.1 Selection of Order of Implementation of Optimization Techniques

From analyzed results, the order of implementation is proposed.

- **Case 1:** Both delay and loss are within tolerable range.
  Guaranteed load service is applied to further enhance it.
- **Case 2:** Delay is tolerable but loss is high.
  Buffer size of access points is increased till acceptable value of delay. Further, RED is applied as the next option. If loss persists, third option is to apply FEC technique. Lastly, controlled load service is applied.
- **Case 3:** Loss is less but delay is high.
  The buffer size of access points is decreased till acceptable value of loss. Weighted RED is applied as the next option with random drop type to ensure fairness. Controlled load service is applied as the last option.
- **Case 4:** Both delay and loss are poor.
  Controlled load service is applied. RED is implemented with small difference between maximum and minimum thresholds as next option.

3.3 Test-Bed Implementation Phase

Our proposed approach is initially implemented in a single call scenario in the test-bed as described in Section 3.1. Network conditions are varied using NEWT. Heuristic categories are described in Table 5. Fig. 5 shows the state-space diagram for the call. Heuristics for each state are shown in Table 6 and the transition function for every link is described in Table 7. The average delay is 120 ms and packet loss is 1%. The average MOS is 3.3. Thus the call is of acceptable quality. Readings from VQManager as seen in Fig. 6(a) suggest that loss and delay remain uniform and tolerable in degraded conditions.

Table 5. Category of Heuristics

Heuristic Category	Description
Excellent	Delay<=100 ms, Loss <=1%, MOS>=4
Good	100ms<Delay<=150 ms,1%<Loss<=2%, 3.5<=MOS<4
Average	150ms<Delay<=180 ms,2%<Loss<=5%, 2<=MOS<3.5
Poor	Delay>180 ms, Loss>5%, MOS< 2

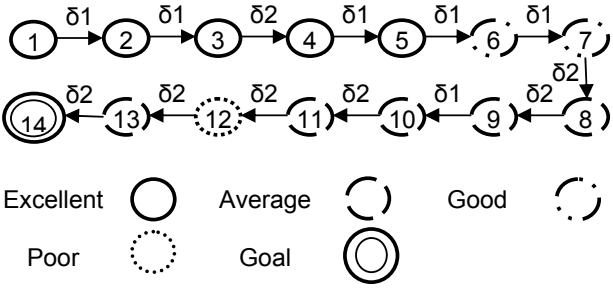


Fig. 5. State transition diagram for the call

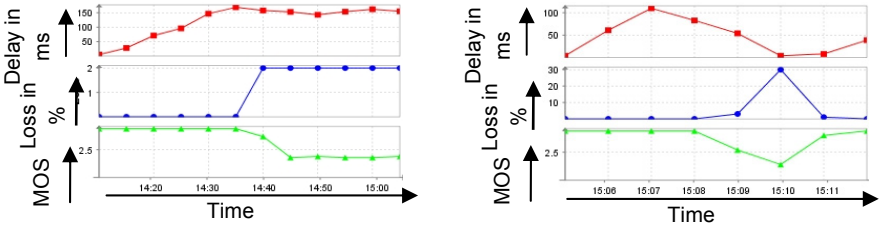
Table 6. Heuristics for each state during the call

State	Delay (ms)	Loss(%)	MOS	Duration (s)	Comments
1	6	0	4.4	1	Start State
2	19	0	4.4	420	Excellent
3	85	0	4.4	530	Excellent
4	69	0	4.4	90	Excellent
5	95	0	4.4	350	Excellent
6	131	0	4.4	137	Good
7	147	0	4.4	126	Good
8	169	0	4.4	600	Average
9	164	0	4.4	187	Average
10	160	2	3.3	143	Average
11	169	2	2	136	Average
12	169	2	1.8	136	Poor
13	143	2	2	340	Average
14	163	2	2	250	Goal State

Table 7. Transition function for every link between the states

Link	Transition Functions	Comments
1-2	50 ms latency(δ1)	
2-3	65 ms latency(δ1)	
3-4	Guaranteed service applied(δ2)	Delay decreases to some extent
4-5	80 ms latency(δ1)	
5-6	120 ms latency(δ1)	
6-7	(δ1)	Delay varies significantly
7-8	Buffer size reduced to 50(δ2)	Delay decreases
8-9	Buffer size reduced to 30(δ2)	Delay reaches threshold mark.
9-10	0.01 Loss rate(δ2)	1 out of every 100 packets is lost.
10-11	Buffer size increased to 45(δ2)	To decrease loss & enhance MOS
11-12	Buffer size increased to 60(δ2)	It is done to decrease increasing loss. MOS now becomes uniform.
12-13	RED applied with max threshold of 100 and min threshold of 50(δ2)	As buffer size cannot be increased further due to increase in delay, the next best ranked action is chosen.
13-14	Controlled load is applied(δ2)	MOS gets improved.

The proposed algorithm is implemented in a multiple call scenario. Two calls are made and mapped as state-space diagrams. Fig. 6(b) shows that delay and loss remain tolerable. The heuristic values are shown in Table 8.

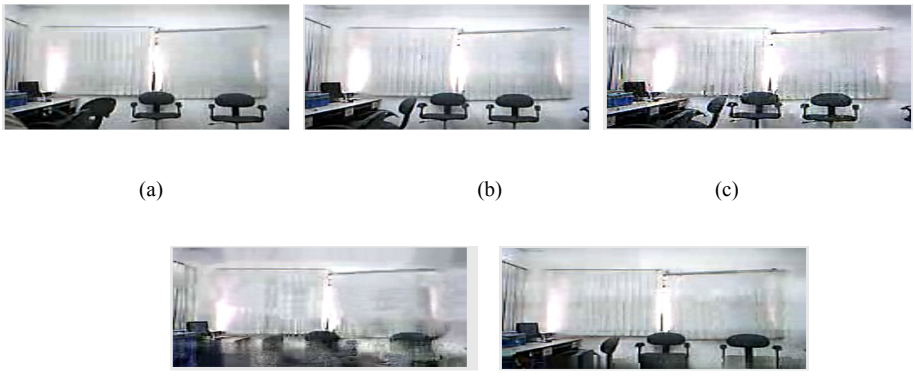


(a) (b)

Fig. 6. Variation of delay, loss and MOS in (a) single and (b) multiple call scenario

Table 8. Heuristic values in multiple call scenario

Parameters	Minimum	Maximum	Average
Delay (ms)	4	110	49
Loss (%)	0	30	5
MOS	1.4	4.4	3.6



(a) (b) (c)

(d) (e)

Fig. 7. Images during a video call with (a) 0 % (b) 3 % (c) 4 % (d) 6 % (e) 5 % loss

Finally, the algorithm is applied to video call which degrades with increasing network loss as seen in Fig. 7. Applying the algorithm makes it recognizable as in Fig. 7(e). Thus QoS of video call is maintained adaptively.

4 Benefits of the Proposed Algorithm

The benefits of the algorithm are discussed hereby. State-space search has been directed towards finding optimal routes between nodes in various network conditions as in [15]. However, little work as in [16] and [17] has been done with respect to mapping network related problem to state-space problem and solving it. Our proposed

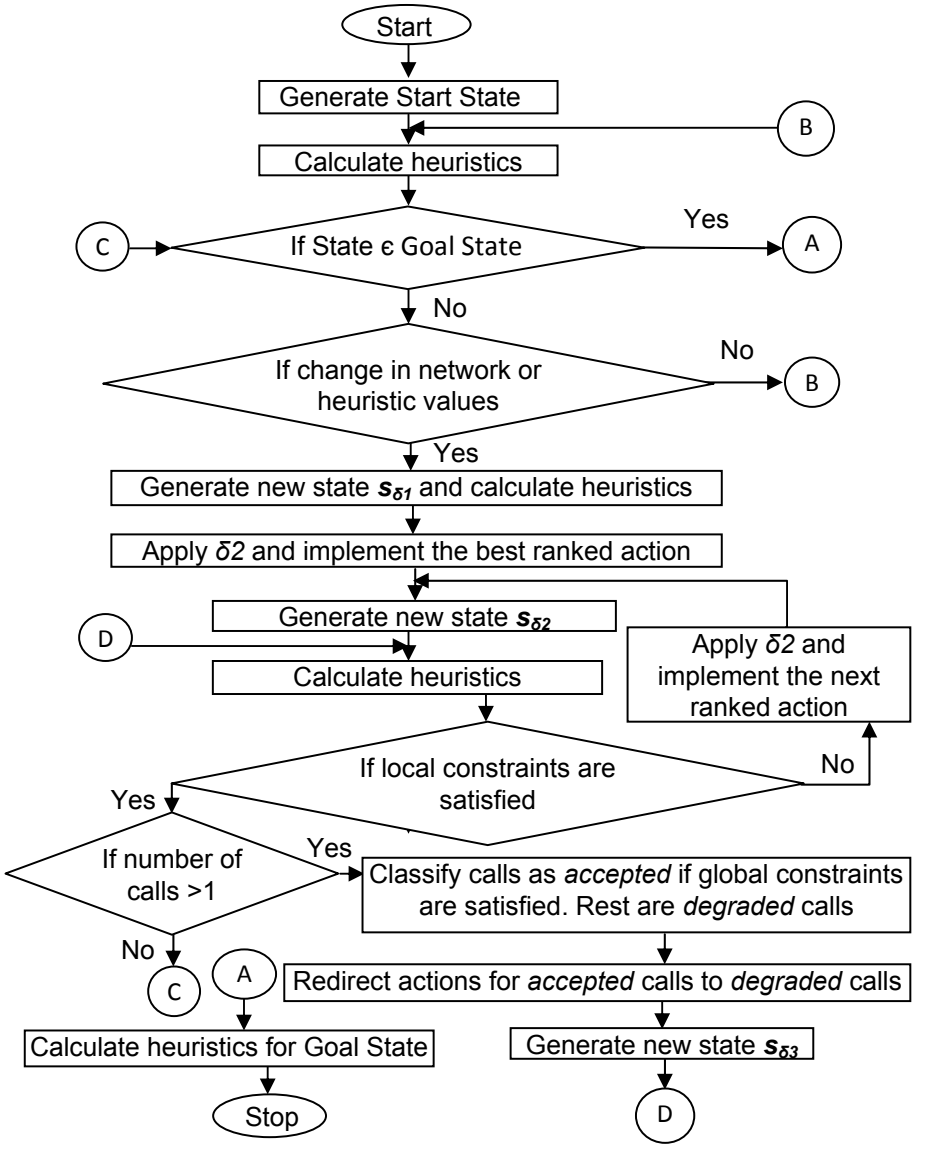


Fig. 8. Flowchart depicting the proposed approach

approach addresses QoS maintenance in real-time systems by mapping it into state-space search problem. This is advantageous as advanced search techniques and new optimizations, for instance as stated in [18] and [19] can be applied. It aims to build automated system which applies transition functions to satisfy constraints, relieving applications from complexities of QoS maintenance and maintaining transparency [20]. From state-space perspective**,** it satisfies inferential adequacy, inferential efficiency and aquisitional efficiency.

5 Conclusion

In this paper, we have dealt with the problem of adaptive QoS maintenance under dynamic and diverse network conditions and applied optimization techniques accordingly. Test-bed readings verify the fact that application of proposed algorithm in single and multiple voice and video calls keeps delay and loss within threshold limits even as network conditions vary with time. The algorithm further ensures that no conflict arises during application of optimization techniques as proper sequence is maintained among them. While VoIP traffic binds this algorithm to real-time heuristic search, modern optimizations in dynamic search domain can be further applied to state space search approach.

Acknowledgements. The authors deeply acknowledge the support from DST, Govt. of India for this work in the form of FIST 2007 Project on "Broadband Wireless Communications" in the Department of ETCE, Jadavpur University.

References

1. Khasnabish, B.: Implementing Voice over IP. Wiley-Interscience, John Wiley & Sons, Inc., Hoboken (2003)
2. May, M., Bolot, J., Diot, C., Lyles, B.: Reasons not to deploy RED. In: Proc. Seventh International Workshop on Quality of Service, pp. 260–262 (1999)
3. Rich, E., Knight, K.: Artificial Intelligence, 2nd edn. McGraw-Hill Science/Engineering/Math, New York (1990)
4. Koenig, S., Likhachev, M., Liu, Y., Furcy, D.: Incremental Heuristic Search in Artificial Intelligence. AI Magazine 25(2), 99–112 (2004)
5. Koenig, S.: Real-Time Heuristic Search: Research Issues. In: International Conference on Artificial Intelligence Planning Systems, Pennsylvania (June 1998)
6. X-Lite, `http://www.counterpath.com/x-lite.html`
7. Brekeke Wiki, `http://wiki.brekeke.com`
8. ManageEngine VQManager manual, `http://www.manageengine.com`
9. NEWT, `http://research.microsoft.com`
10. Chen, J-H., Lee, W., Thyssen, J.: RTP Payload Format for BroadVoice Speech Codecs, RFC 4298 (December 2005)
11. Branden, B., et al.: Recommendations on Queue Management and Congestion Avoidance in the Internet, RFC 2309 (April 1998)

12. Li, B., Hamdi, M., Lang, D., Cao, X., Hou, Y.T.: QoS-Enabled Voice Support in the Next-Generation Internet: Issues, Existing Approaches and Challenges. IEEE Communications Magazine 38(4), 54–61 (2000)
13. Wroclawski, J.: Specification of the Controlled-Load Network Element Service, RFC 2211 (September 1997)
14. Shenker, S., Partridge, C., Guerin, R.: Specification of Guaranteed Quality of Service, RFC 2212 (September 1997)
15. Zhu, T., Xiang, W.: Towards Optimized Routing Approach for Dynamic Shortest Path Selection in Traffic Networks. In: International Conference on Advanced Computer Theory and Engineering, pp. 543–547 (December 20-22, 2008)
16. Mandal, S., Saha, D., Mahanti, A.: Heuristic search techniques for cell to switch assignment in location area planning for cellular networks. In: IEEE International Conference on Communications, vol. 7, pp. 4307–4311 (June 20-24, 2004)
17. Franqueira, V.: Finding Multi-Step Attacks In Computer Networks Using Heuristic Search And Mobile Ambients. Ph.D Dissertation, University of Twente, Netherlands (2009)
18. Chakraborty, T., Mukhopadhyay, A., Misra, I.S., Sanyal, S.K.: Optimization technique for configuring IEEE 802.11b access point parameters to improve VoIP performance. In: 13th International Conference on Computer and Information Technology (ICCIT), pp. 561–566 (December 23-25, 2010)
19. Mukhopadhyay, A., Chakraborty, T., Bhunia, S., Saha Misra, I., Sanyal, S.K.: Study of enhanced VoIP performance under congested wireless network scenarios. In: Third International Conference on Communication Systems and Networks, 2011, pp. 1–7 (January 4-8, 2011)
20. Aurrecoechea, C., Campbell, A.T., Hauw, L.: A survey of QoS architectures. Multimedia Systems 6(3), 138–151 (1998)