SUMMARY Wireless technology has become widely popular and an important means of communication. A key issue in delivering wireless services is the problem of congestion which has an adverse impact on the Quality of Service (QoS), especially timeliness. Although a lot of work has been done in the context of RRM (Radio Resource Management), the delivery of quality service to the end user still remains a challenge. Therefore there is need for a system that provides real-time services to the users through high assurance. We propose an intelligent agent-based approach to guarantee a predefined Service Level Agreement (SLA) with heterogeneous user requirements for appropriate bandwidth allocation in QoS sensitive cellular networks. The proposed system architecture exploits Case Based Reasoning (CBR) technique to handle RRM process of congestion management. The system accomplishes predefined SLA through the use of Retrieval and Adaptation Algorithm based on CBR case library. The proposed intelligent agent architecture gives autonomy to Radio Network Controller (RNC) or Base Station (BS) in accepting, rejecting or buffering a connection request to manage system bandwidth. Instead of simply blocking the connection request as congestion hits the system, different buffering durations are allocated to diverse classes of users based on their SLA. This increases the opportunity of connection establishment and reduces the call blocking rate extensively in changing environment. We carry out simulation of the proposed system that verifies efficient performance for congestion handling. The results also show built-in dynamism of our system to cater for variety of SLA requirements.

**key words:** case based reasoning, congestion control, high assurance, radio resource management, service level agreement, intelligent agents

1. Introduction

Cellular networks alongside mobile phone users have grown at a staggering pace in the last decade. It has taken over as a standard for business communication nowadays thus making the availability of high quality service a pivotal entity for the Service Providers (SPs). With the increase in telecom service providers, changing a SP has only been reduced to a mere “SIM change” in many areas in the world. So failing to deliver the promised service is no more a tolerable option. In a competitive environment like this it is of paramount importance that SP engineers and maintains its network to provide the promised service is no more a tolerable option. In other words, a competitive environment like this it is of paramount importance that the network to provide the promised service is no more a tolerable option. In our work, call blocking rate is the primary focus of interest. Call blocking rate for in-coming connections can increase with rise in Multiple Access Interference (MAI) and overloading of cell causes congestion to break threshold barrier. MAI includes interference that is generated at the BS by all the up-linked signals from its own cell and others [1]. One user’s signal acts as a source of interference for others. The greater the number of users a WCDMA (Wideband Code Division Multiple Access) system has, the greater will be the overall level of interference within the system. Handoff rejection to a mobile terminal moving across the cell boundary may also increase both call blocking and dropping rates [11]. So there is a requirement to introduce an efficient mechanism for Radio Resource Management (RRM). RRM is the system level control of co-channel interference and other radio transmission characteristics in wireless communication systems, for example cellular networks, wireless networks and broadcasting systems [14]. RRM becomes more complicated when providing the flexible, higher bandwidth services, and maintaining the best system capacity in 3G systems like WCDMA. We propose a novel technique in this paper to achieve desirable QoS. In order to keep this proposition in context, first we shall briefly describe the related work.

Typical mobile networks allocate bandwidth as per user requirements. However, allocating bandwidth resource goes wasted while not in use by the customer. Alternatively a better approach is to use a single and relatively smaller “bucket” bandwidth for all prime customers in addition to the normal frequency allocation. In the wake of congestion,
system uses this "bucket" bandwidth to satisfy the needs of prime users only but under normal circumstances all users have same privileges. The difference between the two lies in the atrophy of bandwidth, which is much larger in the former than the latter.

Another conceptual approach proposed in [3], [10] is to exploit CBR as the source for maintaining history of successful cases for future references. CBR is an Artificial Intelligence (AI) problem solving technique that catalogs experience into "cases" and matches the current scenario to the experience to find solution when the system undergoes congestion [2], [4]. There are a number of key issues in this technique. For example, while it proposes the basic flow of control, it does not provide system architecture, details of algorithms and implementation. Moreover, CBR library management along with the retrieval of records at the time of congestion has remained unhandled. Another issue is the size of the CBR library as the bigger library gives better chance of finding right solution but as a downside it takes considerably longer time to retrieve it. On the other hand if the right solution is unavailable in the repository it will have to use adaptation which again is a hiccup to the performance.

In this paper we propose a realizable system architecture and detailed design with CBR to control congestion through efficient RRM. The use of intelligent methods along with efficient call admission gives autonomy to the BS and enables it to dynamically choose a suitable policy for the new network situation and apply it to the system. Therefore, the proposed system architecture assures continuous service provision and utilization on supply-demand basis as per SLA between SP and users based on the concept of Autonomous Decentralized System (ADS) [13], with the following two principles:

- Autonomy: subsystems are self-regulatory, and can manage their behavior.
- Adaptability: system is capable to adapt itself in response to dynamic conditions.

Multi-agent systems by virtue of their subsystem level autonomy and decentralization characteristics facilitate realization of the proposed system with these two principles. The theme is to minimize the probability of occurrence of congestion prior to its happening, and to meet the SLA in autonomous and decentralized way. The rest of the paper is organized as follows. Section 2 gives details on the requirements for SLA. Section 3 presents the proposed system architecture and Sect. 4 describes the details of the proposed CBR based algorithm. Section 5 describes the performance evaluation of the proposed system and Sect. 6 summarizes research work presented in the paper.

2. Requirements for Service Level Agreement

The target system consists of a heterogeneous users division with variable service requirements. This results in dynamic situations for the providers as now the goal is not to provide a uniform array of services and quality but to manage each diverse group's requirements individually. Therefore it is incumbent upon SP to decide a yardstick to measure the provided services to ensure quality service.

The SLA includes the obligation to fulfill the multifaceted requirements of user's heterogeneous groups. It is a contract which identifies the agreed upon services that has to be offered by the SP to the users. It underscores the QoS requirements for each class of users. In our work, the congestion is the prime issue under consideration as this is the most critical problem faced by any network. In this context, a network is said to be suffering from congestion when its requisite SLA is not met. Therefore, it is key recruitment for SLA that the network does not enter into the congestion state. The concept of SLA is the focal point as the entire problem and solution described in the paper revolves around this concept. An example of SLA is to "not let the call blocking rate exceed a threshold 'x' for a particular user class in a specific interval of time" [2]. Call blocking rate means the rate at which the connection requests for a particular class of users are rejected. In other words, the ratio of number of calls blocked for a particular user class to the total number of accepted calls in a given interval of time gives forth the call blocking rate.

\[ \text{Call blocking rate} = \frac{\text{No. of calls blocked}}{\text{Total accepted calls}} \]

We consider three classes of users i.e. Gold, Silver and Bronze with priority descending from Gold to Silver and subsequently lowest to Bronze. Priority in practice is determined by the amount of payment each user class pays. Also, Gold and Silver users are served at the cost of ignoring Bronze users. It has been shown that treating users differently at network access level achieves desirable results in terms of managing system capacity and customer satisfaction [3]. Therefore this paper proposes the use of high assurance techniques [9] to meet heterogeneous and changing requirements for bandwidth allocation in 3G systems to achieve SLA.

3. Proposed System Architecture

The proposed system in essence consists of agents structured hierarchically into three layers as depicted in Fig. 1. The three-layered agent-based architectural style forms a base to ensure autonomy to the BS with each layer accountable for its distinct responsibility. Inspired from the conceptual approach [4], we have proposed a detailed design along with a novel Retrieval and Adaptation Algorithm based on CBR in order to achieve SLA. The impetus to our work is to make available an alternative approach to call handling when no bandwidth is available to cater for further users. The theme is to present a system architecture that is not only graceful in dealing with customers but at the same time reduces congestion in pre-emptive manner. In order to achieve this goal we have exploited CBR technique to propose Case Retrieval and Adaptation Algorithm in this paper. The de-
The detail of the algorithm is described in Sect. 4. The acme of our work is the reduction in time complexity of solution discovery based on proposed architecture, and it also copes with the dynamic network due to changing users’ requirements.

As for the agent layers, from the bottom the first layer is Reactive layer (Fig. 1). The main task of this layer is to handle incoming call, invoke the Call Admission Control (CAC) function to check whether it should be handed the call space or temporarily put in buffer. The call setup utility also functions at this layer. Further details are available in Sect. 3.1.

The local planning agent layer follows suite. It is the “brain” of our system and encompasses the CBR technique in addition to the retrieval and adaptation algorithm working in conjunction with CBR. It facilitates for the queries of Reactive layer. It has been comprehensively discussed in Sect. 3.2.

The cooperative planning agent layer is tasked with carrying out coordinated actions among groups of agents in different cells. Autonomous distributed resource management between cells also takes place in this layer. This layer has only been included for the purpose of completeness [7]. The focus of this paper will, however, remain on the first two agent layers discussed in Sects. 3.1 and 3.2. In the paper we shall use the term agent layer and layer interchangeably.

3.1 Reactive Layer

The reactive agent layer is responsible to handle problems in a quick and responsive manner. The respective components have been shown in Fig. 2. The Reactive layer has to perform its actions in real time so that an RNC (Radio Network Controller) maintains QoS. The actions include assigning the connection requests and executing CAC. CAC mechanism is to make decision about the incoming request channel allocation. Before moving on to its detailed usage in our system, let us first look at the background theory and the mechanism of CAC. It is applied in real time environments and is an important tool in maintaining QoS of the system. It is a preventive congestion control procedure and operates during the Call Setup phase. It is a common fact that as the number of connections increases, the quality of existing connections decreases. Every new connection acts as a source of interference for existing ones. Therefore, a need crops up for a mechanism that can monitor access to the network in order to prevent service degradation of active connections. CAC controls the allocation of network resources to roaming and new users based on their availability and the new call is accepted only if the system has sufficient bandwidth to ensure the fulfillment of QoS requirement for existing callers [6].

As for the mechanism in CAC, we have taken a combination of ideal and Signal-to-Interference Ratio (SIR) based CAC in our work. Ideal admission control ensures that QoS is guaranteed to all active connections when accepting a new request whereas in SIR based CAC, a threshold is set considering all the users including requesting user [8]. QoS attributes are negotiated between each new user and reactive layer at the time of start up. As the SIR decreases, it gets difficult to guarantee QoS requirements. Consequently, we need to set a target value for SIR to keep the QoS at its negotiated level. It should however be noted that SIR remains dynamic during a session varying on the impact of “noise” introduced by each new call. Therefore, it is crucial to limit the maximum number of users at any given instant by keeping the SIR value greater than the threshold. So as soon as the CAC is invoked, it decides whether to accept or reject the request based on the policies of local planning layer. Local planning layer mainly handles the CBR which is deputed with carrying out tasks related to buffer stay time. If the policy recommends the acceptance at CAC, the connection is established and added to the system otherwise the request is sent for buffering.
If the call is sent to buffer, the buffering section identifies the caller's class and checks how much time it can reside in the buffer. The call request remains in buffer as shown in Fig. 2, for a maximum of its defined time. Buffering time for each class may be dynamically changed depending upon the severity of congestion. The more the congestion for a class, the more its users stay in the buffer and have a higher probability to get channel access. Another sub unit called the buffer time decreasing, shown in Fig. 2, decrements time for all mobile clients waiting in the queue (buffer). In the wake of a call termination, new capacity is created and the system offers channels to the mobile in front of the queue. The newly assigned call's entry is also removed from the buffer. If the buffering time for a particular user expires while in the buffer, it will be left out and congestion rate for that class is recalculated. This approach provides a beneficial opportunity to priority users as their calls stand a chance of getting a channel in the wake of congestion.

During the normal functioning of reactive layer, periodical checks of monitoring are carried out by the monitoring module taking into account the current state and checks for any prediction of congestion in the near future, so as to take preemptive measures. For example, we can consider a case whereby the call blocking rate for Gold users approaches 0.03 (the threshold value set in SLA for Gold users as explained in Sect. 4.1). The system would identify the forthcoming congestion and take relevant steps to reduce congestion beforehand. In this way the system is adaptive to cater for the congestion before it actually occurs.

As for the relevant steps of preventive measure, when congestion is alarmed, monitoring module consults CBR which proposes appropriate solution to the current situation to reduce congestion as explained in Sect. 4.1.

The system needs to be aware of the status of calls especially when they are terminated. This objective is achieved through the finished calls component. Its special feature is that it neither uses a loop nor a separate thread for checking the finished calls. The process has been made simpler as the notification job has been delegated to the individual calls which notify the system when it leaves.

The system updation component gets its feedback from finished calls component so when the new channel slot is available, the CAC can offer it to the caller in front of the queue, on priority basis.

3.2 Local Planning Layer

The local planning layer is the "neuro-center" of our system with CBR having its base in this layer. CBR is an AI problem solving technique that catalogs experiences into "cases" and matches the current problem to the experience [5]. The solution finding pursuit is carried out by exploring the successfully solved prior problems, called cases, or adapting old solutions to meet new demands. This approach is beneficial as it continuously keeps the agent in learning and improving state. A good reference to use CBR in controlling mobile networks is described in [4]. CBR seems to be the most fitting approach as different traffic patterns can be studied, and together with the solution they can be indexed in the case library to use for solving future problems. Solution Adaptation (SA), a very critical component, also resides in the planning layer. When CBR fails to provide a satisfactory solution, case adaptation is invoked. Its algorithms need to decide the buffer stay time for different classes of users. It has to be realized that an ideal match is hard to find and in most cases the system will have to use adaptation on nearly matched case. More description is available in Sect. 4.2.

4. CBR Based Congestion Control Algorithm

CBR is the primary component of planning layer. CBR consists of a library which carries with it a set of records to solve the arising issues. It works in conjunction with Case Adaptation Algorithm when required e.g. if library fails to find an exact match.

CBR library contains a case part (current network situation) and a solution part. The later contains the buffer stay time for previously dealt users. SLA in our work is defined in terms of tolerable threshold for call blocking rate, we have to set particular values for threshold for each class. In this paper, we consider three types of users' classes. For Gold users, threshold is assumed 0.03 while for silver users, this limit is 0.06. These thresholds also represent the relative priority of Gold vs. Silver. The Bronze has lowest priority of being assigned a call slot in critical situation, so it has zero buffer stay time and hence excluded from the CBR library. The case library is stored on the physical medium in the form of a binary tree in XML format. The binary format of XML storage has its benefit of providing us with an even better access to cases with fewer instructions [15]. The details of the CBR based Congestion Control Algorithm is given in the Sects. 4.1 and 4.2.

4.1 Case Retrieval Algorithm

There are three main scenarios that can occur for Case Retrieval Algorithm as follows:

- Congestion for Gold users only,
- Congestion for Silver users only,
- Congestion for both Gold and Silver users.

For each of these scenarios, the algorithm looks into the library using different searching preferences explained in 4.1.1 and 4.1.2.

4.1.1 Case Retrieval: Congestion for Gold Users

When the congestion is restricted to gold users only, the algorithm will search on the basis of GoldRate of a solved case in the library because there is no congestion for other users and preference is given to Gold. Upon locating the exact match which is the ideal scenario, it is applied otherwise case adaptation (explained in 4.2) modifies the nearly matched solution and uses it to reduce congestion. It should
however be noted that finding exact match is difficult as the call blocking rates continuously alter and perfect matches are rare. For example, if the perceived network state is as follows:

\[
\text{GoldRate} = 0.035, \quad \text{SilverRate} = 0.035
\]

Now going by the SLA as explained in Sect. 2, this technique will return the record 3 (Fig. 3) as a matched case. As it is an exact match, the searched case is exactly applied to the system.

On the contrary if the system current situation is as follows:

\[
\text{GoldRate} = 0.034, \quad \text{SilverRate} = 0.04
\]

Traversing through the records in library, the system fails to locate an exact match. It means there is no identical match in the library for this congestion situation of the system. Therefore, on the basis of \(D\) (as explained in Sect. 4.1.2, Eq. (1)) the agent retrieves the nearest match record 3 (Fig. 3) and using case adaptation in Sect. 4.2 it modifies the match to be applied to the system.

A situation may arise where two or more records get retrieved. In that case the search also takes the SilverRate as secondary attribute among retrieved matches.

For example if the current congestion situation is:

\[
\text{GoldRate} = 0.031, \quad \text{SilverRate} = 0.04
\]

4.1.2 Case Retrieval: Congestion for Gold and Silver Users

In this case Gold and Silver users both suffer congestion simultaneously and system calculates an absolute distance \(D\) of the current congestion case from the solved cases in the library and chooses the one with least value of \(D\). Distance \(D\) is calculated as follows:

\[
D = G' \times C + S'
\]

where

\[
G' = \text{currentCase.GoldRate} - \text{libCase.GoldRate}
\]
\[
C = \frac{\text{SilverCongestThreshold}}{\text{GoldCongestThreshold}}
\]
\[
S' = \text{currentCase.SilverRate} - \text{libCase.SilverRate}
\]

Also currentCase.GoldRate references the GoldRate in the current case, while libCase.GoldRate references GoldRate in the case retrieved from the library, and same notation is applied to currentCase.SilverRate and libCase.SilverRate.

In Eq. (1), \(D\) is calculated for each case in the library where \(G'\) is the difference of current gold class congestion rate with the gold rate of instance in library under consideration, likewise \(S'\) is the difference of current silver class congestion rate with the silver rate of the same record. These differences are taken into consideration in order to select a record from the library in closest match to the current congestion scenario. The Variable \(C\) in the equation is a constant used to equalize the proportional strength (0.03 for Gold and 0.06 for Silver) of gold and silver rates, so that they have the equal effect on the selection process of a record with minimum \(D\). In our hypothesis according to Sect. 2.5, value of \(C = 2\), because SilverCongestionThreshold = 0.06 and GoldCongestionThreshold = 0.03.

\[
C = 0.06/0.03 = 2
\]

4.2 Case Adaptation Algorithm

When the CBR library fails to provide a satisfactory solution, SA is invoked. It modifies the nearly matched case (solution) retrieved from the CBR library to calculate dynamic time for different classes of users for staying in the buffer to reduce congestion whenever it occurs. Its algorithms need to calculate this buffer stay time using the following equations.

\[
\text{NewGoldBufferTime} = A + (M - X) \times C + (N - Y) \tag{2}
\]
\[
\text{NewSilverBufferTime} = B + (N - Y) \tag{3}
\]

where

\(A\) is described in Sect. 4.1.1
\(A = \text{GoldBufferTime of the nearly matched case retrieved from the CBR}\)
\(B = \text{SilverBufferTime of the nearly matched case retrieved from the CBR}\)
\(X = \text{GoldRate of the nearly matched case retrieved from the} \)
CBR
Y = SilverRate of the nearly matched case retrieved from the CBR
M = GoldRate of the current case (situation) of the system
N = SilverRate of the current case (situation) of the system
These variables are also shown in Fig. 3.

It is obvious that the new buffering times will almost be the same as the CBR retrieved case because it is the nearest match and rest of the difference is interpolated by the Eqs. (2) and (3) above. \((M - X)\) and \((N - Y)\) are the factors due to which current case and CBR retrieved case are not exactly matched. To calculate the NewGoldBufferTime, \(A\) is the most important factor because it is GoldBufferTime of the CBR retrieved case. \((M - X)\) calculates the difference between GoldCongestionRate of current system case and CBR retrieved Case. Similarly, \(B\) is the most important factor while evaluating NewSilverBufferTime. \((N - Y)\) computes the difference between SilverCongestionRate of current system case and CBR retrieved case.

The factors \((M - X)\) and \((N - Y)\) can also be negative depending upon the fact that GoldRate of the current case is less than the one retrieved from the CBR and SilverRate of the current case is less than the one retrieved from the CBR respectively. Depending upon the values of \((M - X)\) and \((N - Y)\), \(A\) and \(B\) are increased or decreased respectively. This algorithm can be elaborated through an example as follows:

Let’s say the current condition of the system is noted as:

\[
\begin{array}{c|c|c|c|c}
\text{GoldRate(M)} & \text{SilverRate(N)} & A & B \\
0.034 & 0.04 & ? & ?
\end{array}
\]

And retrieved nearest case from CBR is record 3 in Fig. 3.

\[
\begin{array}{c|c|c|c|c}
X & Y & A & B \\
0.035 & 0.035 & 3.75 & 2.00
\end{array}
\]

This example shows that the GoldRate of the current case i.e. 0.034 is less than the CBR retrieved case i.e. 0.035, which means that congestion for Gold in current case is less than the record 3 above. So NewGoldTime should be a bit lesser than A, that’s why \((M - X)\) factor is negative to reduce \(A\). If \(M - X\) is positive, it shows that GoldCongestionRate of current case is greater than that of the CBR retrieved case. In such scenario NewGoldBufferTime should be greater than \(A\) so it adds positive factor into \(A\). The more a connection request stays in buffer, the more chances it will have to get channel. Similar logic is applied to calculate NewSilverTime using Eq. (3). \((N - Y)\) in Eq. (2) adds or subtracts from \(A\) because there is only single buffer shown in Fig. 2 in which all the users stay waiting for channel. So their buffer stay time affects one another. As Gold is of the highest priority we add a portion of SilverTime in GoldTime to give it more preference.

5. Performance Evaluation

In this section we describe the performance evaluation of the proposed system. These results present the performance analysis of case retrieval and case adaptation algorithm. The measure of this performance is represented by call blocking rate.

5.1 Test Bed Setup

Simulation of the WCDMA environment has been carried out in Matlab while the proposed algorithms have been implemented in Java and then integrated with Matlab. In the test bed setup, we consider one BS with random incoming calls. Our system could supports almost 500 calls at a time depending upon signal power of the system and flexible support for ratio of Gold, Silver and Bronze users. Traffic load is controlled by changing the inter-arrival time between connection requests. Congestion is simulated by decreasing the inter-arrival time between consecutive call requests. The system performs amicably well until the call counter reaches 500. When numbers of connections increase threshold, the system starts showing signs of congestion. The proposed algorithm is called into action as the call blocking rates exceeds respective thresholds. The respective scenarios of congestion with respect to each user class are discussed next in the Sect. 5.2.

5.2 Congestion Control

We simulated the system with and without applying the proposed architecture. We checked for congestion of Gold and Silver users respectively. Since, there is no SLA requirement for Bronze customers, they can suffer from congestion in the network to any extent. It should be noted that under the congestion scenario Gold and Silver users were served at the cost of Bronze customers.

5.2.1 Congestion Behavior of the WCDMA Traffic without Proposed System

As the simulation was put into action, the connection requests started pouring into the system. The normal behavior system soon turned into a congested one as the gap between subsequent calls reduced. Since there was no policy for buffering, the call blocking rate kept increasing. Gold, Silver and Bronze users’ call blocking rates are shown on the graph in Fig. 4 without considering any type of SLA.

5.2.2 Congestion Behavior of the WCDMA Traffic with Proposed System

Connections arrive in the system randomly. Also the arrival of Gold or Silver users is unpredictable. Results show that the threshold barrier is never broken when policy is established. As mentioned in Sect. 4.1, the threshold for gold customers is 0.03 and for silver 0.06. So, as the blocking rate reaches at 0.025 for gold or 0.055 for silver connections, the case retrieval and case adaptation algorithms are activated as to cater for the congestion before it occurs. The threshold values we assumed were not hard and can be
changed as per SLA.

At point P which is the threshold value for Gold user, the notifications went up and system applied new policy. As a result, congestion dropped continuously (Fig. 5) as connection requests got buffering time which increased their chances of getting requests fulfilled. The key point is to prevent congestion before it attacks the system.

Similarly for Silver connections, the retrieval and solution adaptation algorithm is activated when Silver blocking rate approaches 0.06. The blocking rate started dropping at point Q after new policy is applied and the system assure the SLA (Fig. 6). Figure 6 also depicts the fact that SLA for both Gold and Silver users is marinated.

5.2.3 Congestion Behavior of the WCDMA Traffic with Proposed System with Modified SLA

In this section, we analyze our system with varied SLAs as the system is dynamic and able to comply with customized situations and requirements autonomously. In our variation of SLA, we took the Gold threshold value as 0.04 and for Silver as 0.075. Figures 7 and 8 depict the condition in the system as simulation progresses along.
The congestion for gold later tends to touch the threshold boundaries at Point P (Fig. 7), similarly for Silver, it tends to touch at P and Q (Fig. 8). At these points gold and silver respectively are on the brink of breaking thresholds which are duly handled by the system in autonomous and decentralized way.

In all the figures (Figs. 4–8), we see a hiatus between the beginning of simulation time and the congestion lines depicting the state of congestion for each class. This interval is the period of non-congestion when all the requests are duly complied with in a normal way. Once the congestion tends to approach the threshold rate, the system applies the policy to ensure congestion free operation with respect to SLA of users’ class. Simulation confirms the validity of the proposed system architecture and algorithm to achieve QoS as per SLA in dynamic cellular network environment.

6. Conclusions

In this paper we have proposed a novel congestion control technique which guarantees QoS through high assurance for users’ heterogeneous and changing requirements. The hallmark of proposed system is the SLA that serves as an important benchmark in the system primarily to maintain QoS for the customer various classes. The proposed intelligent agent-based system architecture is realized with CBR in conjunction with proposed retrieval and adaptation algorithm that act as a source for SLA based congestion avoidance through usage of case library. We carried out evaluation of the proposed system using Matlab for network environment and Java as implementation language for the proposed system architecture and algorithm. The system performs very well when applied to a number of congestion scenarios, and ensures QoS for various classes of users as per SLA in dynamic environment. Through the proposed system, 3G networks can combat congestion efficiently in order to achieve the high standards of QoS which will eventually help SPs retain and win new customers in dynamic environments. The future direction is to extend the autonomous RRM among adjacent cells in cellular networks.

References


Hafiz Farooq Ahmad is Specialist Engineer in Communication Technologies (Comtec), Sendai, Japan. He received Ph.D. from Tokyo Institute of Technology in 2002. He also has joint appointment as Associate Professor at School of Electrical Engineering and Computer Sciences (SEECS), National University of Sciences and Technology (NUST), Rawalpindi Pakistan. His main research topics are autonomous decentralized systems, multi-agent systems, autonomous semantic grid, semantic web and formal methods for software engineering. He is a member of IEEE.

Hiroki Suguri is a Vice President and Chief Technology Officer with Communication Technologies, which he co-founded in 2000. He received Ph.D. from Iwate Prefectural University in 2004. His main research topics are multi-agent systems, grid services and semantic web. He is a member of IPSJ, IEEE and ACM.
Muhammad Qaisar Choudhary received BS degree in Information Technology from National University of Sciences and Technology (NUST), Pakistan in 2006. He started his research career as a research assistant in NUST and served for one year. He later proceeded to Royal Institute of Technology (KTH), Sweden for pursuing his MS in Distributed Systems. His research interests include distributed algorithm development, semantic web, machine learning and artificial intelligence.

Ammar Hassan is a graduate of NUST Institute of Information Technology (Now SEECS). After his graduation he joined the 3G Research group at SEECS as a researcher where he worked in areas related to Call Admission Control (CAC), SIP and Congestion Control. Currently he is a student of KTH (Sweden) doing his MS with specialization in Software Engineering of Distributed Systems (SEDS). He has been actively participating in challenging projects related to Distributed Systems and is keenly working on multi-agent platforms and intends to pursue his research in this same domain.

Ali Liaqat received M. Sc. degree in applied Mathematics from the University of the Punjab, Lahore, Pakistan, in 1989, Master of Science in engineering from Quaid-i-Azam University, Islamabad, Pakistan, in 1992 and the Master of Engineering from the University of Electro-Communications, Tokyo, Japan, in 1999. He received the degree of Doctor of Engineering from the University of Electro-Communications, Tokyo, Japan in 2003. He conducted cooperative research as a postdoctoral fellow at the University of Tokyo, Japan from 2003 to 2005 under the JSPS postdoctoral fellowship program. Currently, he is working as a Principal Scientist at PAEC, Pakistan. He is a member of the Japan Society for the Promotion of Science. His research interests include neural networks, artificial intelligence, pattern recognition, inverse problems. He received a Silver Medal from the University of the Punjab, Lahore, Pakistan in 1992, the Best Presentation award at the World Conf. on 21st Century Maths., GCU, Lahore, Pakistan in 2004.

Muhammad Umer Khan received BS degree in Information Technology from National University of Sciences and Technology (NUST), Pakistan in 2006. His research interests include distributed computing, intelligent congestion control, semantic web, artificial intelligence, and sensor networks.