SLA-Driven Flexible Bandwidth Reservation Negotiation Schemes for QoS Aware IP Networks

David Chieng[1], Alan Marshall[2], and Gerard Parr[3]

[1] Faculty of Engineering, Multimedia University,
63100 Cyberjaya, Selangor D.E., Malaysia.
htchieng@mmu.edu.my
[2] School of Electronic & Electrical Engineering, The Queen's University of Belfast,
Ashby Bld, Stranmillis Rd, BT9 5AH Belfast, Northern Ireland, UK.
a.marshall@ee.qub.ac.uk
[3] School of Computing and Information Engineering, University of Ulster,
BT52 1SA Coleraine, Northern Ireland, UK.
gp.parr@ulster.ac.uk

Abstract. We present a generic Service Level Agreement (SLA)-driven service provisioning architecture, which enables dynamic and flexible bandwidth reservation schemes on a per-user or a per-application basis. Various session level SLA negotiation schemes involving bandwidth allocation, service start time and service duration parameters are introduced and analysed. The results show that these negotiation schemes can be utilised for the benefits of both end user and network provide such as getting the highest individual SLA optimisation in terms of Quality of Service (QoS) and price. A prototype based on an industrial agent platform has also been built to demonstrate the negotiation scenario and this is presented and discussed.

1 Introduction

In today's complex network environment, QoS provisioning for real-time applications over IP-based networks is a great challenge. Firstly, service and network providers will have to deal with a myriad of user requests that come with diverse QoS or Service Level Agreement (SLA) requirements. The providers will then need to make sure that these requirements can be delivered accordingly. To address these issues, we propose a unique Service Level Agreement (SLA)-driven service provisioning architecture that enables flexible and quantitative SLA negotiations for network services. In this paper we focus on bandwidth reservation and management on a per-user, per-application or per-flow basis. Software agents are employed to assist the service provider in guiding, deciphering and responding quickly and effectively to users' requests. These satisfy the two most important performance aspects in SLA provisioning; availability and responsiveness.

This paper is organised as follows: Section 2 first presents the service provisioning architecture and a set of associated SLA parameters. Section 3 and 4 introduce the respective SLA utilities and prototype system. Section 5 presents the SLA negotiation schemes enabled by the proposed architecture and section 6 describes the simulation environment. The simulation results and evaluations are then discussed in section 7. Finally the related work, the overall analysis and conclusions are drawn in section 8 and 9 respectively.

2 SLA Driven Service provisioning Architecture

In general, many end users/customers and service or network providers are still unable to specify SLAs in a way that benefits both parties. For example, the network providers may experience service degradation by accepting more traffic than their networks can handle. On the contrary, they may fail to provide services to the best of their networks' capabilities. In this work, bandwidth reservation is emphasized since it is the single most important factor that affects the QoS. The limits for delay, jitter and buffer size can be determined by the bandwidth reserved for a flow [1]. The architecture not only provides immediate reservation, it also allows bandwidth resource to be reserved in advance. The high-level service parameters that can be negotiated are summarized in table 1 as follow:

Table 1. SLA Parameters

SLA Parameters	Description
Price (*P*)	Maximum price for this connection per transaction or price per unit bandwidth ($/*b*).
Start Time (*Ts*)	Reservation start time or activation time.
Session Length (*T*)	Reservation session duration or reservation enforcement duration.
Guaranteed BW (*b*)	The amount of bandwidth guaranteed/reserved.
Option (Ω)	Priorities setting and preferences

The generic architecture as shown in figure 1 is proposed [2][1]. The Domain Manager (DM) generally manages the network domain. It communicates with the policy server that administrates policies, rules and actions for different services stored in a policy repository. In these policies, various limits such as maximum bandwidth that a user can reserve at a time, maximum or minimum reservation duration, etc, can be specified and enforced through other elements in our architecture. The Path Table (PT) stores the logical 'reservable' path or route ID info. The Resource (bandwidth) Reservation Table (RRT) comes in the form of a resource/time table. This allows the network provider to lookup and allocate network resources (bandwidth) at present and also in the future. The User Service Database (USD) stores individual customer's SLA information such as service ID, the respective bandwidth allocation (*b*), agreed service activation time (*Ts*), session duration (*T*), path ID or routing option, billing option, and also other rules and policies bound to this particular user. After an SLA is

[1] The generic architecture shown in figure 1 has been presented in [2]. Current paper provides a more refined architecture and extended [2] with a prototype implementation and some additional results

accepted, the DM maps the required configurations down to the Policy Enforcement (PEP) layer where the appropriate QoS control is performed.

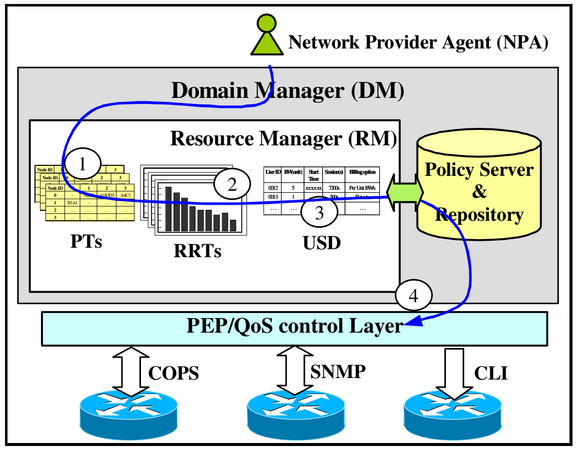


Fig. 1. Generic SLA Architecture

The policies can be enforced in such a way that when the duration of resource reservation T expires, the connection will automatically revert to best effort mode. Alternatively, the allocated bandwidth will be sustained until it is needed by other incoming non-preemptable sessions. The advance reservation facilities are desirable for services are such as Video on Demand (VoD) and news broadcast where service start time are known in advance. In a situation where the desired session time cannot be specified or it is not known a priori such as in IP voice calls, an alternative scheme is necessary. Agents can be employed to optimise resource usage bilaterally such as performing dynamic resource negotiations. Here, autonomous agents also play an important role in enhancing service discovery process i.e. via advertisement. This is essential in today's heterogeneous network environments where not all networks offer services with QoS options. Various admission control schemes or policies can then be enforced to control and optimise resources, and at the same time maximising user satisfaction and network provider's profit.

3 SLA Utilities

We extend the SLA management utility model proposed by [3]. It gives both end user and network provider a unified and computationally feasible approach to perform session request/admission control, quality selection/adaptation, resource request/allocation decisions, etc. In this work, a user request i for a guaranteed service session can be represented by:

$$u_i\ (b_i,\ Ts_i,\ T_i,\ P_i,\ \Omega_i) \qquad (1)$$

After translating to a resource usage function in terms of reserved bandwidth, this can be represented by:

$$r_i\,(b_i,\, Ts_i,\, T_i) \tag{2}$$

In order to make the request more elastic, tolerances or variation levels are introduced and so that the probability of a request being rejected can be reduced. For example, a video subscriber might not mind waiting for another 1 or 2 minutes until the requested resources become available. Alternatively, if this is not possible, the subscriber may opt for a slightly lower video quality at some discounted price. The service provider may on the other hand propose an alternative if the demanded service cannot be granted.

Occasionally, the provider may want to propose a higher quality higher bandwidth video session with a cheaper price in order to maximize network utilization. On another occasion, a provider may propose a higher bandwidth with shorter session rather than lower bandwidth with longer session. The utility can be represented by equation (3) as follows:

$$u_i\left(b_i \pm a, Ts_i \pm b, T_i \pm c, P_i \pm d\right) \tag{3}$$

Where a, b, c and d are the tolerance limits acceptable by the user. These tolerance parameters can be embedded as part of user policies or preferences (Ω) as shown in equation (1).

4 System Prototype

To demonstrate the agent-enhanced SLA brokering and negotiation scenarios, a system prototype that consists of a real agent system environment has been developed. More detailed case studies and implementations can be found in [4]. In this work, we used the Fujitsu Phoenix Open Agent Mediator (OAM) platform to build the service provisioning system. The goal was to develop highly flexible, robust and dynamically extensible distributed network service broker prototype. In this platform, agents are realised using Phoenix servlets that communicate via HTTP over TCP/IP. Agent servlets are invoked by responding to HTTP calls from browsers or other agent servlets.

Phoenix OAM introduces a distributed mediation facility where the agents' execution flows can be arranged dynamically. With this facility, an agent is known by its "functions" i.e. offered services rather than its URI. This information is registered at a mediation table through advertisement. Hence, multiple servlet agents registered under the same "function" can be accessed in parallel. In addition, servlet agents can be dynamically loaded and unloaded at run-time (service plug and play can be realized). When a mediator agent receives a request, it will find the desired agents by collaborating with other mediators that reside in other network domains.

The prototype demo illustrates the SLA brokering and negotiation scenario. Here, the network stakeholders involved are the end user, the access service provider, the network/connectivity provider and some content providers. When a User Agent (UA) is first invoked by an end-user, it downloads a page from the Access Service Provider

Agent (ASPA). The page offers a range of network services as shown in figure 2. The ASPA will broker the request to the target Content Provider Agent (CPA) that represents the VoD service provider. The agent then replies to the user with a VoD service subscription page as shown in figure 3. The subscription page allows the user to select the movie title, the desired quality, the desired movie start time, the tolerance parameters, the maximum price he or she is willing to pay, etc.

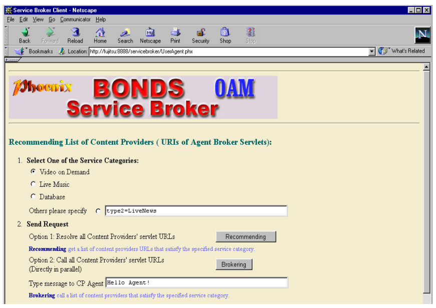


Fig. 2. Service Brokering Page

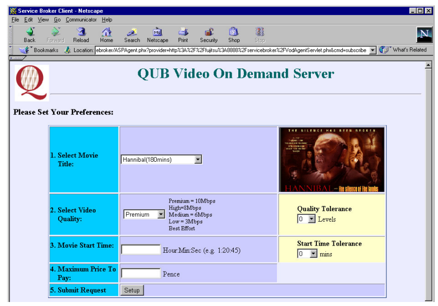
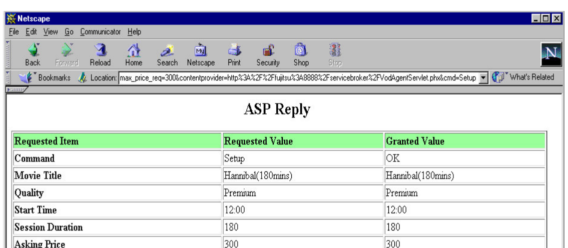


Fig. 3. VoD Service Subscription Page

Assuming the user wants to watch the movie "Hannibal" that requires a duration of 180mins; the desired video quality is "Premium"; the desired movie start time is at 13:00; and the maximum acceptable price for this service is 300 pence. The maximum quality tolerance is 1 level, i.e. from premium (10 Mbps, q_i) to high (8 Mbps, q_{i-1}) if the requested quality cannot be provided. Alternatively, if the requested quality cannot be honoured at the specified start time, the user may be willing to accept a tolerance of maximum ±5 minutes. The resulting SLA utility is therefore:

$$u_i\left(b_i = 10 - 2, Ts_i = 1300 \pm 5, T_i = 180, P_i = 300\max\right)$$

Once the "Setup" button is clicked, the UA then proceeds to encode these preferences into Phoenix's URLEncoded parameters and sends it to the ASPA. In order to setup a network connection with the required SLA, the ASPA needs to negotiate with the Network Provider Agent (NPA). If the requested SLA can be satisfied, the service will be granted.



Requested Item	Requested Value	Granted Value
Command	Setup	OK
Movie Title	Hannibal(180mins)	Hannibal(180mins)
Quality	Premium	Premium
Start Time	12:00	12:00
Session Duration	180	180
Asking Price	300	300

Fig. 4. Reply from ASPA

Figure 4 shows the reply if all the requested SLA parameters are granted. If the requested SLA (with the tolerances) cannot be honoured, the ASPA may issue a "Reject" or initiate a new phase of negotiation.

5 SLA Negotiation Schemes

With the proposed service provisioning architecture, dynamic SLA negotiations can take place i.e. between a User Agent (UA) and a Network Provider Agent (NPA). We introduce four novel session level SLA negotiation schemes i.e. *Bandwidth Negotiation at Resource Limit (BNRL)*, *Guaranteed Session Duration Negotiation for both Session Cut Short (SDN-CS)* and *Temporary Session Bandwidth Drop Off (SDN-TBD)*, and *Guaranteed Session Start Time Negotiation with Delay (STN-D)*. The impact of these negotiation schemes on service availability, network utilisation or revenues and mean user satisfaction are analysed. In this work, three SLA performance metrics are introduced.

Rejection Probability (p_{rej}). This parameter directly reflects the service availability. It is vital for most service/network providers to maintain a minimum level of service availability as part of their SLAs. We define the overall rejection probability as:

$$\rho_{rej} = \frac{N_{rej}}{N_{rec}} \tag{4}$$

Where N_{req} = Total number of SLA requests rejected and N_{rec}= Total number of SLA requests received.

Percentage Mean Utilisation or Mean Reservation Load ($\%\overline{R}_s$). It is defined as the percentage of total mean reserved bandwidth in Resource Reservation Table (RRT) in relative to its total bandwidth capacity (C). This is represented by:

$$\%\overline{R}_s = \frac{\left(\frac{\sum_{t=t_1}^{t_2} R(t)}{t_2 - t_1 + \tau}\right)}{C} \cdot 100\% = \frac{\overline{R_{T_{col}}}}{C} .100\% \tag{5}$$

Where $t = t_1, t_1 + \tau, t_1 + 2\tau, \ldots t_2$ and $t_1 \le t \le t_2$. $R(t)$ is the reservation load in the RRT at minimum 'reservable' timeslot t, $\%\overline{R}_s$ is the mean RRT utilisation measured during a period of time, T_{col}. $T_{col} = t_2 - t_1 + \tau$ where t_1 (inclusive) is the start collecting data period, and t_2 is the stop collecting data period. $\%\overline{R}_s$ reflects the revenue earned if a usage-based billing such as in GPRS is adopted.

User Satisfaction Index. It is impossible to keep all the users happy all the time. Sometimes, the NPA has to reject some requests or negotiate the users' SLA requirements. In this study, a parameter called User Satisfaction Index is introduced to represent a user's satisfaction. It is defined as the ratio of what a user is granted to what the user originally requested. This can be represented with a generic function as follows:

$$\text{Index} = \frac{\theta_{granted}}{\theta_{requested}} \tag{6}$$

The NPA needs to ensure that the average user satisfaction index does not fall below a certain threshold.

6 Simulation Environment

A network model using Block Oriented Network Simulator (BONeS) [5] has been developed to study the negotiation schemes above. The logical capacity of the links in terms of 'reservable' bandwidth across the network is fixed at C bps. The minimum and maximum limits for bandwidth requested per user (b_r) are set 1 and 156 units respectively where one unit may represent 64kbps. The offered load $\%\overline{R}_q$ (same definition as $\%\overline{R}_s$) is defined as percentage mean load requested in relative to the total bandwidth capacity of the link (*C*). The b_r distribution profile with the above parameters is shown in figure 5. The idea is to create a distribution that generates more requests for lower bandwidth, i.e. voice calls. The average b_r for this distribution is measured at 58.32 units or 3.73Mbps if 1 unit = 64kbps.

Figure 6 was generated by scanning through the RRT at a particular instance with *C* set at 1562 units and with 20% of incoming requests were requesting bandwidth resource in advance. It is shown that at *t*=5000s, some future resources have already been reserved.

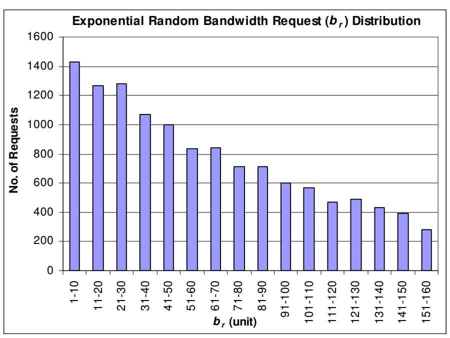


Fig. 5. b_r Request Distribution

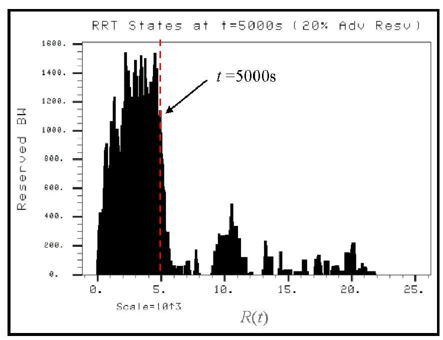


Fig. 6. 20% Advance Reservation

7 SLA Negotiation Schemes Evaluation

The following experiments were simulated over 200,000s or 55.55 hours simulation time. To ensure the simulation is in a stable state, t_1 is set at 100,000s and t_2 is set at 200,000s. In the following experiments, each simulation was run using different random number generator seeds in order to investigate the deviation caused by the simulation model. The data are then used to plot the confidence intervals i.e. mean, maximum and minimum values of the results. Due to the limited space however, only the results from the three session level SLA negotiation schemes i.e. *Guaranteed*

Session Duration Negotiation for both Session Cut Short (SDN-CS) and *Temporary Session Bandwidth Drop Off (SDN-TBD),* and *Guaranteed Session Start Time Negotiation with Delay (STN-D)* and are presented in this paper. The results for *Bandwidth Negotiation at Resource Limit (BNRL)* negotiation scheme can be found in [2].

7.1 Guaranteed Session Duration Negotiation

This section investigates the scenario when the end users are willing to negotiate the guaranteed session duration. Logically, this is only applicable to those applications whose bandwidth does not need to be guaranteed throughout the session such as web browsing, FTP, or other less critical real-time applications. Here, two schemes are proposed namely the *Session Duration Negotiation with Session Cut Short* and *Session Duration Negotiation with Temporary Session Bandwidth Drop Off.*

Session Duration Negotiation - Cut Short (SDN-CS)

In this scheme, if the requested session T_r (from Ts to $Ts+T_r$) is not available, the maximum available 'continuous' session duration will be proposed to the end users. A user's duration tolerance T_{tol} is defined as the percentage of T_r when the bandwidth is not guaranteed or $T_{tol} = \left(\frac{T_r - T_g}{T_r}\right) * 100\%$, where T_g is the guaranteed session duration granted. The service request utility function with session duration negotiation can be represented by $u_i(b_i, Ts_i, T_i - T_{tol} * T_i, P_i)$. This only happens if the required bandwidth has already been booked by other Advanced Reservation (AR) calls. In other words, without AR calls, session negotiation will not happen. An additional policy is applied here where only the immediate is allowed to negotiate session duration. This is a fair assumption as AR calls are unlikely to negotiate session duration although it is also possible. In the following experiments, only a small percentage of AR calls are considered i.e. 0%, 10%, 20% and 30%. 0% AR means that all the incoming calls are requesting for immediate reservation. It is worth mentioning at this point that all the following experiments were carried out at $90\,\%\overline{R}_q$.

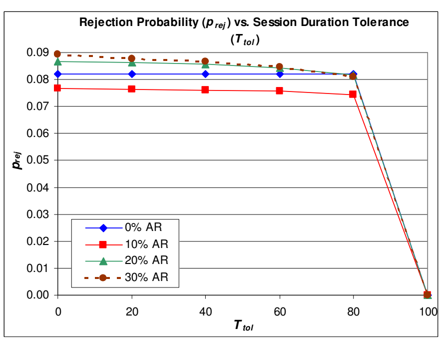


Fig. 7. ρ_{rej} vs. T_{tol}

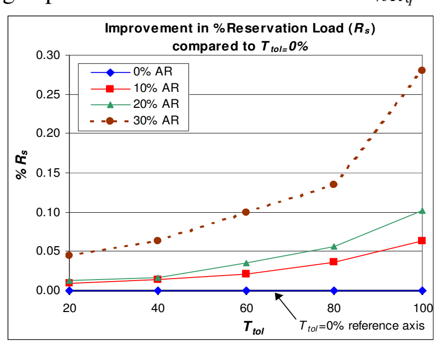


Fig. 8. $\%\overline{R}_s$ Improvement with T_{tol}

From figure 7, generally the drop in p_{rej} is not significant from T_{tol} = 0% to T_{tol} = 80%. For 0% AR, T_{tol} has no effect on p_{rej} since no negotiations take place. When there is 10% AR calls, the users experience lower p_{rej}. This is because the chance of requests being blocked by prescheduled AR calls is low. However p_{rej} increases when there are 20% and 30 % of AR calls because more immediate reservation calls are being blocked by the existing AR sessions. The chance of a prescheduled AR session blocking new AR calls is negligible. In fact none has been recorded for 20% and 30% AR.

Figure 8 shows that $\%\overline{R}_s$ generally improves if session duration is negotiable (compared to 0%T_{tol}). The effect is more significant if the percentage of AR calls is high. However the degree of improvement is quite small with this scheme (<0.3 $\%\overline{R}_s$ even with 100% T_{tol}). It also proves that without AR call, SDN-CS basically has no effect on $\%\overline{R}_s$. In terms of revenue, if 1 Mbps of guaranteed bandwidth is priced at is \$2 per hour, only around \$0.10 extra revenue per hour is earned with T_{tol} = 20% at AR calls=30%.

In this experiment, the Session Duration Index (SDI) is used to represent overall users' satisfaction. It is defined as the mean ratio of session duration granted over session duration requested or mean (T_g/T_s). The simulation result shows that the difference in SDI is almost negligible. The reason is very few negotiations are actually successful with this scheme. Therefore, an alternative session negotiation scheme was introduced.

Session Duration Negotiation - Temporary Bandwidth Drop Off (SDN-TBD)

In this scheme, rather than having the guaranteed session duration being cut short, the users may be willing to tolerate intermittence drop off in bandwidth. This is used if the total duration or total number of time slots τ (T_u), when b_r cannot be granted, is not larger than the tolerance level, T_{tol}. Therefore T_{tol} =(T_u/T_r)*100% for this scheme. However at each T_u time slot, the maximum available bandwidth will be granted. In other words, 'Best Effort Reservation' is performed at each T_u time slot. In this experiment, $\%\overline{R}_q$ is also set at 90% but the percentage of AR calls is fixed at 20%. Figures 9-11 compare the two schemes (SDN-CU and SDN-TBD).

As shown in figure 9, SDN-TBD scheme suffers a higher p_{rej} when T_{tol} is <50% but yields a lower p_{rej} when T_{tol} exceeds 50%. This is because $\%\overline{R}_s$ improves significantly with SDN-TBD scheme even with small T_{tol} (figure 10).

The sudden increase in RRT utilisation leaves less bandwidth for future requests and therefore causes p_{rej} to increase suddenly. p_{rej} can only be reduced by further increasing the T_{tol}. These results are expected since SDN-TBD scheme offers 'best effort reservation' when the requested bandwidth at certain RRT time slots are not available. Whereas SDN-CS scheme just provides the session guarantee up until the instance when the first 'bandwidth not available' time slot at RRT is encountered. In terms of revenue, even with only 20% T_{tol}, SDN-TBD yields an extra ~5.6 $\%\overline{R}_s$ as compared to SDN-CS. This is extremely significant as if, for example, 1 Mbps of guaranteed bandwidth is priced at \$2 per hour, then this represents an increase of ~\$11.20 per hour for a 100 Mbps link.

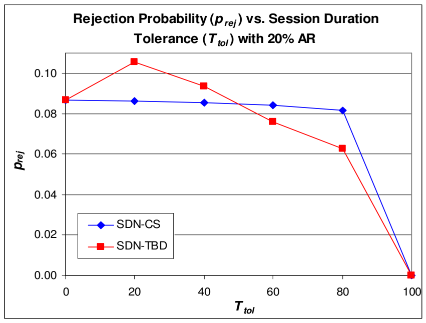


Fig. 9. ρ_{rej} vs. T_{tol}

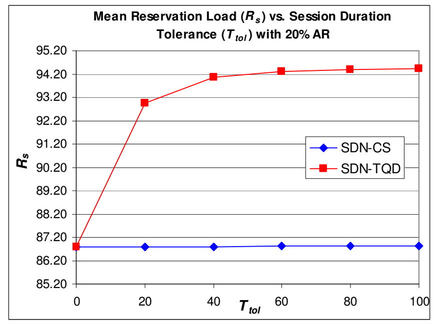


Fig. 10. $\%\overline{R}_s$ Improvement vs. T_{tol}

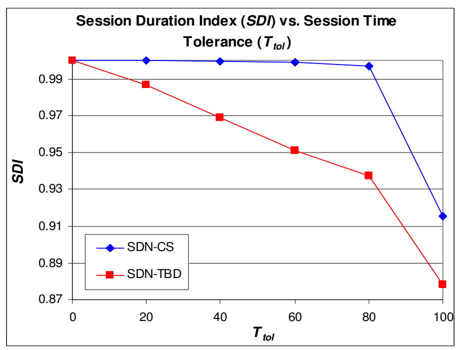


Fig. 11. SDI vs T_{tol}

Figure 11 compares the SDI between these two schemes. For SDN-TBD, SDI is defined as mean $[(T_r - T_u)/T_r]$. Here, SDN-TBD drops significantly with T_{tol} as compared to SDN-CS. This is expected as more negotiations have been taking place.

7.2 Guaranteed Start-time Negotiation with Delay (STN-D)

This scheme can be applied if the user does not mind delaying the guaranteed service start time (Ts) if the requested bandwidth is not available at the desired service or session start time, Ts_r. Rather than asking the user to request again in the future, a network provider can allocate an alternative session start time that is within the user's tolerance limit, Ts_{tol}. Ts_{tol} is defined in unit(s) of time slot, τ (where τ is 1 second). This can be represented by the service request utility function, $u_i(b_i, Ts_i + Ts_{tol}, T_i, P_i)$. In this experiment, different percentages of the advance reservation (AR) calls are considered and the offered load, $\%\overline{R}_q$ is fixed at 90%:

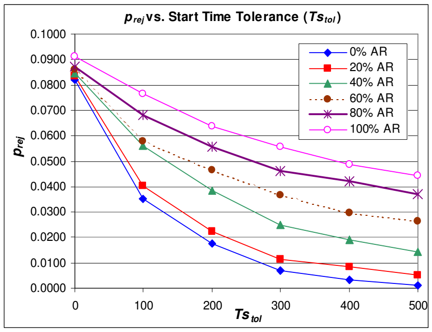


Fig. 12. ρ_{rej} vs. Ts_{tol}

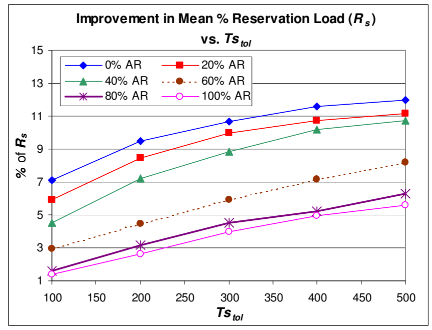


Fig. 13. $\%\overline{R}_s$ Improvement vs. Ts_{tol}

The STN-D scheme produces a significant drop in p_{rej} as shown in figure 12. The effect is less significant if the percentage of AR calls is high because the incoming IR and AR requests are likely to be blocked by other prescheduled AR sessions. Figure 13 shows that $\%\overline{R}_s$ generally improves when session start time is negotiable. The improvement is most significant when all the calls are requesting for immediate reservation or 0%AR. Here, extra 7.08 $\%\overline{R}_s$ is obtained with 0%AR series when Ts_{tol} =100s. From the figure, it can also be deduced that the $\%\overline{R}_s$ improvement drops as the percentage of AR calls increases. This is due to higher blocking probability experienced by new immediate and AR calls if the number of prescheduled sessions is high.

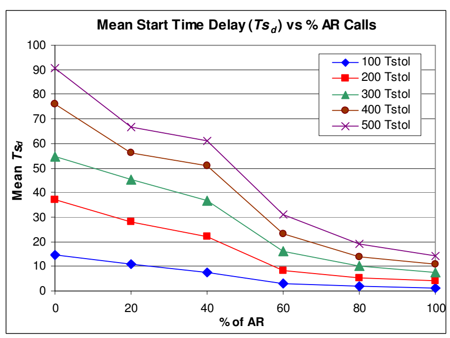


Fig. 14. Mean Ts_d (in second) vs. % AR calls

Figure 14 shows that in general, users experience shorter mean start time delay $\overline{T}_{sd}$ as the percentage of AR calls increases. This is because the session start time of the AR calls falls within a given range, $Ts_{min} \leq Ts \leq Ts_{max}$. Therefore these calls have higher chances to get their desired Ts. In a way this figure also corresponds to the users' satisfaction level where the longer the $\overline{T}_{sd}$, the lower the satisfaction. On average, $\overline{T}_{sd}$ stays lower than 15s when Ts_{tol} = 100s.

8 Related Work

The frameworks for specifying and managing policies related to programmable networks [6] and DiffServ-based network [7] are complementary to our work. The proposed SLA notations, object-oriented methodologies, conflict management methods, etc, can be adopted to enhance our architecture.

The concept of resource reservation in advance has also been addressed in [8], [9], [10], [11], etc. To our knowledge however, none of the above work provides a detailed analysis on session level SLA negotiations. Work by [11] focuses on the design, implementation and evaluation of their Resource Reservation in Advance (ReRA) mechanism by extending the existing RSVP protocol on ATM. The authors also address best-match alternative reservation scenarios similar to that offered by our SLA negotiation schemes. However, in the paper no experimental work was presented

[12] proposes an agent-based reservation system for immediate and AR calls. In their work, a call 'lookahead' time is applied to decide the admission of immediate reservation calls. The effects on rejection probability, pre-emption probability and overall RRT utilisation are studied. We extend their work by looking into session-level negotiation issues involving guaranteed bandwidth, session duration and session start time.

9 Overall Analysis and Conclusions

This paper has presented an SLA-driven service provisioning architecture that facilitates quantitative bandwidth, session duration, session start time preferences negotiations, etc on a per user, per application or per flow basis via SLA. Four novel session level SLA negotiations schemes based on this architecture have been evaluated. The results show that these schemes can be exploited for the benefits of both negotiating parties such as getting the highest individual SLA optimisation in terms of Quality of Service (QoS) and price. It is shown that in most cases, negotiation reduces rejection probability and improves mean RRT utilisation and therefore network's revenues. The choice of scheme to be applied depends very much on the type of applications, the user's preferences and also the load of the link during the time of negotiation. It also depends on the network provider's strategies or policies whether to maximise reservation load (RRT utilisation) or to maximise service availability. Various policies can also be applied to control the session duration, session start time (Ts_{min} or Ts_{max}), etc. Pricing strategies can also be applied to control the users' behaviours. Indirectly, these are seen as a means to manage bandwidth resource [14].

Acknowledgement

The authors gratefully acknowledge Fujitsu Telecommunications Europe Ltd for funding this work and Fujitsu Teamware Finland for the software support.

References

1. Q. Ma, P. Steenkiste: Quality of Service Routing for Traffic with Performance Guarantees, IFIP International Workshop on Quality of Service (IWQoS'97), New York, May (1997), 115-126
2. David Chieng, Alan Marshall: A Policy-Based Bandwidth Resource Provisioning Architecture, IFIP/IEEE Net-Con'2002, Paris, FRANCE, 23-25 Oct (2002)
3. S. Khan, K. F. Li, E. G. Manning: The Utility Model for Adaptive Multimedia Systems, International Conference on Multimedia Modelling, Singapore, Nov (1997)
4. David Chieng, Alan Marshall: Dynamic Network Service Brokering with Open Agent Mediators, International Symposium on Information and Communications Technologies (M2USIC) 2003, Petaling Jaya, Malaysia., 2 - 3 Oct (2003)
5. BONeS DESIGNER Ver 4.01, Alta GroupTM of Cadence Design Systems, Inc
6. Morris Sloman, Emil Lupu: Policy Specification for Programmable Networks, Proceedings of First International Working Conference on Active Networks (IWAN'99), Berlin, June (1999)
7. L. Lymberopoulos, E. Lupu, M. Sloman: An Adaptive Policy Based Management Framework for Differentiated Services Networks, Proceedings of 3rd IEEE Workshop on Policies for Distributed Systems and Networks (Policy 2002), Monterey, California, June (2002).
8. M. Karsten, N. Beries, L. Wolf, R. Steinmetz: A Policy-Based Service Specification for Resource Reservation in Advance, Proceedings of the International Conference on Computer Communications (ICCC'99), Tokyo, Japan, Sept (1999), 82-88
9. D. Ferrari, A. Gupta, G. Ventre: Distributed Advance Reservation of Real-Time Connections, NOSSDAV'95, New Hampshire, USA, April (1995). (Springer Verlag LNCS Vol.1018)
10. L. Wolf, R. Steinmetz: Concepts for Resource Reservation in Advance, Special Issue of the Journal of Multimedia Tools and Applications on The State of The Art in Multimedia, Vol. 4, No. 3, May (1997)
11. Alexander Schill, Frank Breiter, Sabine Kuhn: Design and Evaluation of an Advance Reservation Protocol on top of RSVP, IFIP 4th International Conference on Broadband Communications, Stuttgart, March (1998) 23-24
12. O. Schelen, S. Pink: Resource sharing in advance reservation agents, Journal of High Speed Networks: Special issue on Multimedia Networking, Vol. 7, No. 3-4, (1998) 213-218
13. David Chieng, Alan Marshall, Ivan Ho, Gerald Parr: A Mobile Agent Brokering Environment for The Future Open Network Marketplace, Seventh International Conference On Intelligence in Services and Networks (IS&N2000), Athens, 23-25 Feb (2000), (Springer Verlag LNCS Vol. 1774) 3-15
14. David Chieng, Alan Marshall, Ivan Ho, Gerald Parr: Agent-Enhanced Dynamic Service Level Agreement In Future Network Environments, *IFIP/IEEE MMNS 2001*, Chicago, 29 Oct - 1 Nov (2001), (Springer Verlag LNCS Vol. 2216) 299-312