Subscription Admission Control for End-to-End QoS Multimedia Content Delivery in Multi-domain Environment

Eugen Borcoci, Mihai Stanciu

Universitatea Politehnica Bucuresti (UPB), Romania

Eugen.Borcoci@elcom.pub.ro
Mihai.Stanciu @elcom.pub.ro

Abstract[1]

The framework of this paper is the delivery of multimedia content to many end users, in real time, with end to end (E2E) quality of services (QoS) guarantees through IP multi-domain networks. The paper proposes and studies an admission control (AC) algorithm performed at service management level in each domain manager, in order to establish multi-domain aggregated QoS enabled pipes. Aggregation offers scalability in multi-domain environment, but also one can guarantee E2E QoS for individual flows. The pipes scopes are from multimedia content servers access points, towards the regions where potential users are located, and they will transport many individual flows. The pipes timelife is medium-long term; they can be redefined in some provisioning cycles for network dimensioning. Currently, an implementation of the proposed solution is in progress

1. Introduction

One objective of next generation networks is to offer end to end (E2E) services, QoS guaranteed, for audio, video, voice and multimedia flows. This is still an open issue in multi-domain environment comprising large domains (scalability). The business models considered here involves entities such as: *Service Providers (SP), Content Providers (CP), Network Providers (NP), Customers (CST) (e.g. Content Consumers - CC), Access Network Providers (AN).* A basic set used is: SPs, CPs, NPs and CCs.

This work deals with transport of multimedia content from CPs content servers (CS), through several domains up to regions where potential users (CCs) are located. A *management system* exists in each entity (SP, CP, NP, CC, etc.), [3], [4], [7], [8], [9], composed of *service management (SM)* and *resource management (RM),* the latter including the *traffic engineering (TE)*. The SM deals with service offering to customers - transport independent and TE manages and controls the intra and inter-domain resources, optimising their usage but offering desired level of QoS to the media flows.

One scaleable approach [4], [5], [7], [8], [9], is to establish at SM level, multi-domain logical aggregated pipes, each belonging to a given QoS class, (we assume a few number well known QoS classes) based on *Service Level Agreement/ Specification (pSLA/pSLS)* contracts between providers. Each pSLS request contains [3], [4], [7], [8], all QoS parameters desired and necessary bandwidth. The pipes are setup in advance (based on forecasted data), with respect to real media flow transfer. Their scopes are from CSs access points up to regions where potential users are located. After their logical setup, the pipes are installed in the network (supported by DiffServ and/or MPLS) and advertised to the users. Then the aggregated capacities are "sold" in retail manner, to many customers, through individual contracts *customer-SLA/SLS (cSLA/SLS)* between the SP and each interested CST. The final goal is to assure for each individual flow, the desired set of QoS E2E guarantees. The solution avoids per flow signalling this in inter-domain but uses it at the edges. This approach is adopted in this study.

The aggregated pipes time-life is medium or long term; they can be redefined in some resource provisioning cycles, (RPC) [3], [4], for network dimensioning. The RPC denotes the time period to adjust the anticipated demand and network availability estimates. At RPC epochs the new dimensioning of the network resources (in terms of new traffic trunks (TT) is performed by the RM and availability results (per TTs) are delivered to SM.

A QoS domain manager makes AC (at SM level) of upward requests in order to build a new segment

---

[1] This work has been supported partially by the FP5 IST 507613 Project - EuroNGI and FP5 IST 507637 Project - ENTHRONE

(crossing this domain) of an aggregated pipe. It knows the current availability of intra and inter-domain resources (delivered by RM) and should preserve the QoS level of existing pipes.

The paper is focused on such AC algorithms. Starting from previous other approaches, it is proposed and studied a modified AC, flexible and scalable, policy driven and having a simple implementation. This AC solution is currently implemented in the framework of a research european project.

The paper is organized as follows. Section 2 briefly discusses some previous and current related work. Section 3 introduces the service management farmework and main ideas of this AC. Possible extensions are discussed in Section 4. Open issues and conclusions are presented in section 6.

2. Admission Control Related Work

Many extensive research reports [4], [5], [8], [21] contain sections on AC issues or they are studies dedicated to AC [12] - [22], because of AC crucial role w.r.t. service guarantees and network resources management/control. A lot of solutions centralized or distributed, static or dynamic, more or less adaptive has been proposed in papers and different research projects.

The AC approaches can be coarsely classified, based on the main method used to take decisions about admission/rejection of the new requests for resources. The work [22] identifies several basic solutions for AC, based on: *a priori traffic knowledge/descriptors (model based AC); measurements upon the actual resources utilization; probe packets sent into the network to test its current capabilities (endpoint AC).* Combined methods can be also used.

The *a priori-based AC* supposes knowledge of traffic sources type (model) that are used or will be used in every link and knowledge on the current number of established service instances. The AC also knows the total network resource capabilities in terms of traffic trunks. Based on this information, when a new request arrives, the AC computes the total amount of resources (e.g bandwidth) required (due to previous requests accepted), updates it with the amount corresponding to the new request and will only accept the new request if the minimum updated amount of bandwidth required is less then the available service rate. The actual performance of this method depends essentially on the accuracy of traffic descriptors (note that no measurements of the actual traffic are used) and the degree of conformance of the real traffic flows w.r.t the descriptors. The implementation is simpler than for other methods because it does not involve the monitoring system of the network.

The measurement based AC (MB-AC) does not take its decision based on the user issued information on traffic descriptors, but on information delivered by the network monitoring system. This subsystem makes real-time measurements, thus trying to “learn” the traffic characteristics. Therefore the total demand is not calculated by AC based on traffic models and number of active service instances but it uses the real traffic trunk load value which has been is measured. This method has advantage that the user-specified traffic descriptors can be very simple, which can be easily policed (e.g. peak rate only). The over-provisioning is less probable than in the first method. Also, by measuring the aggregated flows, the statistical values computed are more accurate than estimating the statistical characteristics for individual flow. The main problems of the method are related to the accuracy of measurements (estimation errors), system dynamics and memory related issues, [12].

In the probe-based AC, the end host/application sends probe packets through the network to test the desired path. Using some predefined metric the host decides if the flow can be admitted. The route followed by the probes should be the same for real packets. The probe-based schemes, deduce the network ability to sustain the offered load directly, without relying on pre-allocated network capacity information. They introduce latency in response times, and have inherent problems caused by probes stealing bandwidth from established flows and denial of service when simultaneous attempts congest the network and none is accepted although resources are available), 0.

Combined methods can be used, for instance the method, [322].

3. Subscription Admission Control Framework

This paper considers the *a priori (*model based) pSLS AC which supposes knowledge of traffic sources type (model) generating load for a pipe. The AC also knows the total network resource capabilities in terms of *traffic trunks* (a traffic trunk (TT) is a QoS-class plus a topological scope specified as: *ingress, egress, QoS-class).* When a new request arrives, the AC computes the total amount of bandwidth required (due to previous requests accepted), updates it with the amount of the new request and will say “yes” if the minimum updated amount of bandwidth required is less then the available service rate. The actual performance depends essentially on the accuracy of traffic descriptors (no measurements of the actual traffic are used) and the degree of conformance of the real traffic flows w.r.t the descriptors. The AC performance can be enhanced if considering monitored data on the network.

We consider the *cascaded peering* model of domains which is more scalable [4], [7], [8], [9], [10] than other possible models: hub or centralised. ; each NP should discuss (at SM level) with its direct neighbours only. We will consider here only mono-directional pipes, built from chain of segments, at the request of the SP which is addressing its *pSLS request* to first NP, closest to the CP servers (bi-directional pipes can be built in a similarly); this NP discusses with its neighbour and so on, up to the pipe destination. The pSLS requests are subject to admission control (AC) in the SM of each downward domain. The SP does not need to know the inter-domain topology and routes. The NPs knows the intra and interdomain QoS enabled routes. Finding the routes is out of scope of this paper. A flexible and scalable approach of SM, applicable in large domains, is a two level SM, [3], [5], [8], [9], [10], i.e. service *subscription* and *invocation*. We focus on *service subscription* aspects. The *service subscription* is initiated by SP; it means the process of establishing pSLSs between all pair entities along the path of a pipe. The pSLS *service invocation* means the actual request for installing the resources in the network, offering dynamicity to the solution.

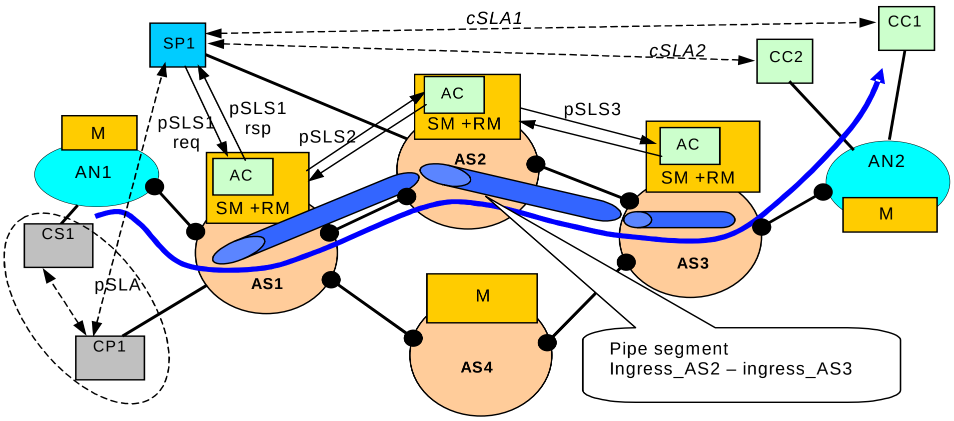

Figure 1: The SLA/SLS contracts in service management

Figure 1 shows an example of a multi domain, having four ASes and business entities SP, CP, CS, CC, AN. Using signaling between managers (SM), the pSLAs (CP1-SP1), pSLSs (SP1-NP1, NP1-NP2, and NP2-NP3) and cSLAs (CC1-SP1, CC2-SP1, etc.) are established. Eventually the digital item from CS1 (owned by CP1) is delivered at request to CC1 (shown in Figure 1), CC2, etc., when the service is invoked by each CC. The AN is considered separately from the rest of the AS chain, because AN technology diversity and management. Controlling QoS in the access network segment is out of scope of this paper.

The *Admission Control* (AC) for pSLS subscription should: keep the QoS level for admitted flows, satisfy the new requests and optimise the network resource utilization. The AC pSLS decisions are based on: initial network availability (intra and inter-domain), as a result of network dimensioning at the beginning of a RPC; current level of aggregated traffic demand; current level of resources for already admitted pipes; traffic description for new requests; policy information for each domain.

4. Subscription Admission Control Algorithm

This section introduce the main ideas of AC used inter-domain to admit/reject the pSLS requests. In general, the AC may verify several QoS realated parameters (delay, loss, jitter). We limit this study to bandwidth allocation. The level of guarantees offered by the AC are also dependent on the methods to share the capacity between several flows. The AC is defined to accept/reject requests for different QoS classes per traffic trunks.

The *QoS-class* here denotes a specific set of transport capabilities that can be supported by the AS network, specified as: *OA (ordered aggregate), delay-bound, loss-bound, [jitter related bounds]*.

The *Ordered Aggregate* has the measning of DiffSErv Technology: the Per Hop Behaviour (PHB) Group with which the packets of a class are treated (possible values: *EF, AF1, AF2, AF3, AF4*).

The semantics of a *Traffic Trunk (TT)*is: *a QoS-class plus a topological scope* (the basic traffic trunk can be is pipe type but generally a TT can also be hose, etc.), specified as: *ingress, egress, QoS-class.* The TTs are aggregates of traffic having the transfer characteristics of the associated QoS class between specific network edges.

The *Resource Provisioning Cycle (RPC):* the time period to adjust the anticipated demand and network availability estimates. RPC is a long period of time. At RPC epochs the new dimensioning of the network resources (in terms of new TTs) is performed by the RM and availability results (per TTs) are delivered to Service management.

4.1 Scope of AC

The AC in ENTHRONE architecture is performed by each domain manager and the target is to accept/reject the request which is done for a uni-directional pipe from starting from an ingress router of this domain up to a destination egress router (located in this domain or or in a remoteone). The responsibility of the local mnanger is to offer a pipe from its ingress up to the egress of the next domain (if this domain is a transit one), as presented in the Figure 2.

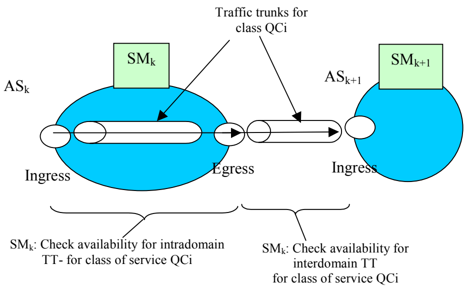

Figure 2: Checking the resource availability by the pSLS-AC-S

4.2 Multiplexing and sharing the bandwidth

The decision of AC are depending on the method to share the available resources. Figure 3 presents an example of different styles of allocating resources. Suppose the case of a (logical or physical) link, having the capacity C provisioned to transport three TTs, (belonging to the same QoS class) while applying one of the four methods of multiplexing.

a Hard reservation.: the total available capacity C is divided between three traffic trunks. In this case one can assure that 100% of resources allocated to each trunk will be always available for this trunk in any network conditions. In fact this is the old telecom method of fixed resource allocation (channels) for an application session. As an example for the trunk TT1 we have $Rhmin1= Ravg1= Rmax1,$ where these are the values of minimum, average and maximum resources allocated to TT1.

b. Hard reservation plus soft partition: we allocate hard for each TT a certain capacity, plus additional amount of resources used in soft partitioned reservation mode: each trunk has a hard reserved portion of resources, but there is a partitioned zone in which each trunk TTi may use a quantity Di of resources, with statistical guarantees only. One can use more (see the gray shadowed zones in the partitioned zone) in certain limits. For instance we have for the trunk TT1: $Ravg1 = Rhmin1 + D1.$ But in fact TT1 may benefit of more bandwidth than Ravg1 due to statistical multiplexing of the flows on the three trunks. Actually TT1 may benefit in some intervals of time from an average bandwidth

$$Ravg1' = Ravg1(1 + g) \geq Ravg1$$

where the factor $g \geq 0$ can be equivalent to a multiplexing gain. The guarantees that can be offered that TT1 will have these resources available will be statistical only. In case of congestion, the network control mechanisms will constraint each trunk TT_i to respect its limit $Ravg_i$ agreed in the contract.

c. Hard reservation plus a shared amount of resources: any TTi can use not only the hard reserved portion of resources but they can share the difference $C- \sum Rhmin_i$ with weaker guarantees than in case b or even no guarantees.

d. Hard reservation plus partition plus a shared amount of resources: this is the most complex method in which each TT: has a hard reserved capacity, a soft partition (which is still for its own and guaranteed in case of congestion) but can also benefit for some sharing pool of resources without guarantees. In conclusion, the *Rhmin* is the first availability limit (hard guaranteed per TT at any time), *Ravg* is the second (available bandwidth for this TT guaranteed by the network at congestion times) and *Rmax* is the final limit (maximum available bandwidth for this TT), with no guarantees because this bandwidth is shared by other TTs.

Note that we can extend the method d. if we allow that the shared portion may be used by TTs belonging *to different* QoS classes (eliminating the restriction that the QoS "planes " are completely disjoint in terms of resources) then one can obtain a better resource utilization. The level of guarantees for the portion of shared resources will depend on the details on how AC is done and how the policing and scheduling parameters are applied. A simple approximation would be to suppose “best effort” service for this part of resources.

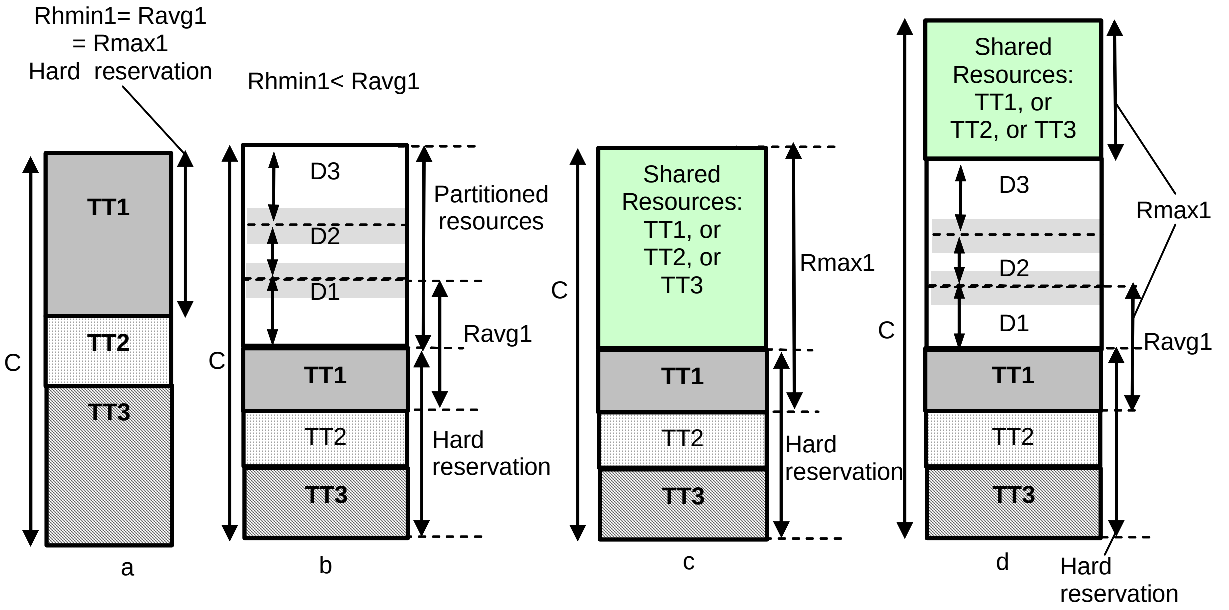

Figure 3 : Multiplexing and sharing the bandwidth

4.3 AC Data Structures and Decision Limits

The AC solution proposed in this paper has partial similarity with ideas of AC presented in [4], [7], while we think that is more flexible, easy adaptable to policy based management and easy to implement. The main information necessary for AC are:

Load demand information: CD_TD_T *- Current Domain Total Traffic Demand* matrix with entries : $\{TT,\ CD\_TD_{Tm},\ CD\_TD_{TM}\}$. The CD_TD_{Tm} and CD_TD_{TM} are respectively the minimum and maximum anticipated demand, for the trunk TT, derived by the newly accepted subscriptions, made within RPC, calculated by multiplexing the minimum/maximum service rates necessary, SR_m /SR_M, for the new pSLS, with the existing load amount (here a simple arithmetical addition can be done or more sophisticated procedures taking into account stistical multiplexing).

A similar matrix is used for inter-domain output trunk between this domain and a neighbour.

Outgoing trunk load demand information: CI_TD_T*: Current Inter-domain Total Traffic Demand matrix* anticipated to be requested by the total subscriptions, accepted within the current RPC and during previous RPCs. This is the traffic that flows through outgoing trunks of this domain.

The format of this data structure is: $\{\mathrm{TT},\ \mathrm{CI\_TD}_{Tm},\ \mathrm{CI\_TD}_{TM}\}$, where: $\mathrm{CI\_TD}_{Tm}$, $\mathrm{CI\_TD}_{TM}$ have similar semantics to $\mathrm{CD\_TD}_{Tm}$, $\mathrm{CD\_TD}_{TM}$ but applied for outgoing traffic of this domain.

Resource availability information: DTT_RAM - Domain Total Trunk Resource Availability Matrix of an AS_i is defined per each unidirectional TT belonging to different QoS class. Its entries are:

$$DTT\_RAM_{i\ C}(k,\ j) = (Rhmin,\ Ravg,\ Rmax\ Del\_min,\ Del\_max,\ ...),$$

where *C* is a given QoS class, *R* is rate (measured in bandwidth units), *Del* is delay, etc. A similar structure gives the availability for inter-domain trunks. Topological information for traffic trunks ends are also available. All these information are delivered by RM to SM at the beginning instant of each RPC. The three values $Rhmin \le Ravg \le Rmax$, can characterise the trunk available bandwidth bound values, assured by the domain RM, respectively: *Rhmin* is hard guaranteed: *Ravg* is only statistically guaranteed (software partioned) even in congestion case; Rmax is non-guaranteed because the *Rmax-Ravg* portion of bandwidth is shared with another trunks. A simplified solution uses only *Ravg* and *Rmax.*

We define an aditional *Overall Quality Level (OQL)* parameter, OQL ∈ [0, 1], to measure the degree of quality/satisfaction that the SP chooses to give to its subscriptions. OQL can be defined per QoS Class (or even per TT). For a given SLS, the OQL is a measure of the confidence level with which the SLS will receive the agreed QoS. OQL = 0 means that, in cases of congestion, no guarantees can be provided for ensuring QoS. OQL = 1 means that even in case of congestion, the SLS will receive and see the promised QoS at the rate value SR_M. The OQL values can be adjusted between RPC cycles, by policies. We define two upper bounds for the resources, which are both some monotonic decreasing functions:

$$UB_M(OQL): [0, 1] \rightarrow [\, Rmin, \infty),$$

and

$$UB_m(OQL): [0, 1] \rightarrow [\, Rmin, Rmax]$$

The analytical expression of UB_M and UB_m functions is for further study and subject for policies.Here *Rmin* can be seen as *Ravg* (but could be also *Rhmin* if we want a simpler approach). The analytical expression of these upper bound functions are quantitative issues and is for further study. Depending on the slope of these functions the bandwidth usage efficiency versus OQL can be lower or higher.

Five cases of pSLS requests are depicted in Figure 4 corresponding to the five QoS classes QC1,..QC5. In each case the values TD_{TM} and TD_{Tm} are computed and compared respectively with UB_M and UB_m. The requests for which the bounds are violated are rejected (simple case) - as in example QC4, or maybe re-negotiated (by proposing to the requester lower values than it requested).

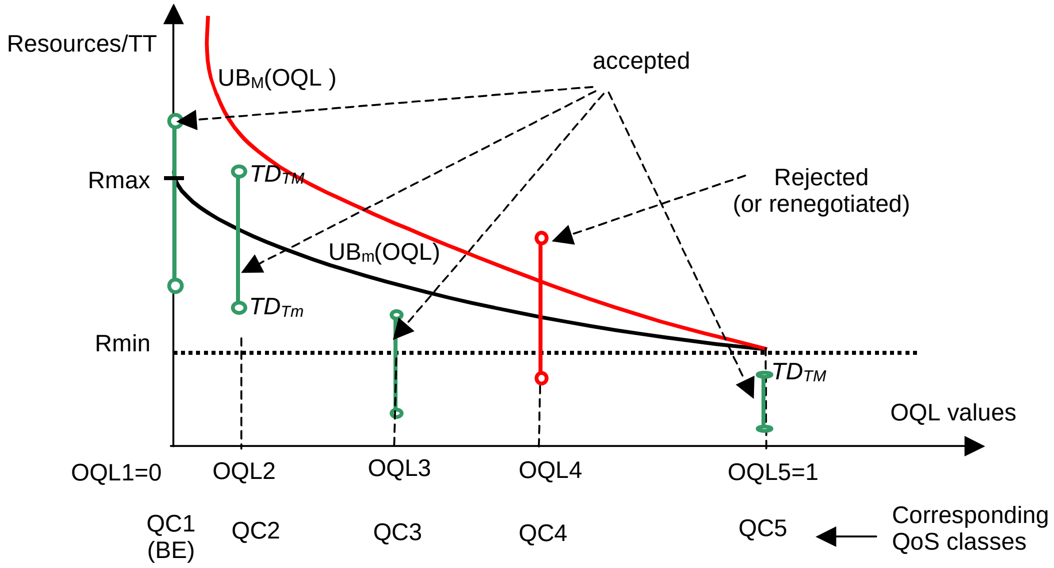

Figure 4: pSLS-AC-S algorithm operation principle

4.4 AC Operation

This sub-section presents a simplified view of AC operation. Each pSLS request request belongs to a given QoS class. When receiving a new pSLS request the AC block performs the following actions:

- determine if the request is locally terminated
- if yes, then determine the TT belonging to the QoS class desired
- adds the traffic of the new pSLS request, supposed to flow through an internal traffic trunk of this domain, to the current domain total demand traffic for the trunk wanted (note that this may be not simply an addition; here policies or some multiplexing gain considerations can intervene)
- updates (temporary) the total traffic demand for that trunk
- compares the total traffic demand with *DTT_RAM* data, available for the TT required
- admits/rejects the request and depending on the of the decision updates the total demand matrix.

If the request is not locally terminated then similar actions are performed for the local trunk. Then the outgoing trunk is determined and next downward domain. The availability of resources is also determined for the outgoing trunk and in case of positive result the request is prolonged to the downward domain. After getting response from downward domain, if it is positive, then a new check is performed for local and outgoing trunks. If still available resources, then a positive answer is returned to the requester. The details of these actions are described in [8].

4.5 Local trunk availability checking

The algorithm to check the local trunk availability checking is summarized below. We consider a domain AS_k. The AS_k traffic trunks resources for the QoS class of services QC_n are expressed as entries in the DTT_*RAM* matrix in the generic format:

$DTT\_RAM_{kn}\ (i, j) = (UB_m(OQL), UB_M(OQL), DLYmin, DLYmax, JITTERRmax, LOSS\ )$

The detailed descriptions of these parameters are given in [ ]. Here the values of $UB_m(OQL)$and $UB_M(OQL)$ are the minimum bandwidth and maximum bandwidth upper bounds defined in the Section 3.4. They have values corresponding to the class QC_n, which in its turn has assigned (by policy) a given value of OQL.

Notes:

1. To simplify the writing we do not specify below the *k* and *n* indexes.
2. Temporary vectors $CD\_TD_{Tm}\_tmp$, $CD\_TD_{TM}\_tmp$ are used as intermediate variables.
3. JITTER and LOSS are not processed in this example.
4. It is supposed that the traffic trunk to be checked is TT_{ij} where the *i* and *j* denote the ends of the TT.
5. Suppose that the bandwidth calculation is done by merely adding the total current bandwidth used with the new request. More elaborate methods can be used (see the next section dedicated to pSLS invocation.

Example: we consider the following request:
$pSLS\_request\_for\_\ TT_{ij}(\ Bmin, Bmax, DLYmax, JITTERmax, LOSS)$ of QoS class *n:*

Actions:

```
$CD\_TD_{Tm}\_tmp(i,j) = CD\_TD_{Tm}(i,j);$ /*total current min bandwidth consumed*/
$CD\_TD_{TM}\_tmp(i,j) = CD\_TD_{TM}(i,j);$
$Success\_flag = 1;$
$Bm\_req = CD\_TD_{Tm}(i,j)\_tmp.Bmin + Bmin;$ /*total min bandwidth request*/
$BM\_req = CD\_TD_{Tm}(i,j)\_tmp.Bmax + Bmax;$
$B\_avail\_min= DTT\_RAM\ (i, j).UB_m;$ /*total min bandwidth available*/
$B\_avail\_max= DTT\_RAM\ (i, j).UB_M;$
If [ $(Bm\_req > B\_avail\_min)$ or $(BM\_req > B\_avail\_max)$]
        Then { $Success\_flag = 0;$ return $(Success\_flag);$}
$D= DTT\_RAM\ (i, j).DLYmax;$ /*temp var for available delay */
If  $(DLYmax\ >D)$
        Then $Drest= DLYmax - d;$
        Else { $Success\_flag = 0;$ exit( );}
Return($Success\_flag, Drest$ );
```

A similar algorithm is applied for the outgoing trunk as for the intra-domain trunk described in the section above with exception that:

- the *Inter-domain Total Trunk Resource Availability Matrix (ITT_RAM)* is considered- containing the outgoing trunks availability
- the *Current Inter-domain Total Traffic Demand matrix* (CI_TD_T) - to give the current total traffic demand on each outgoing trunk
- the *DELAY* parameter value in the request has been adjusted after performing the local check: the delay consumed on the intra-domain trunk has been subtracted, therefore the DELAY parameter of the request represent the maximum admitted delay from the egress router of this domain up to the destination point.

5. Applying policies for pSLS-AC-S

The pSLS-AC-I can be policy driven in order to flexibly adjust the AC process to actual network conditions. In particularl, the analytical expression of $UB_m(OQL)$ and $UB_M(OQL)$ can be subject of policy variation depending on the NP management decision. Therefore the *actual observed* level of OQL (by the end user) for the same relative value of OQL assigned to a given QoS class will depend on the slope of these curves.

The Figure 5 shows two cases:

a. the NP applies a more restrictive (conservative) policy, accepting less multiplexing gain and lower average network resource utilization but offering stronger guarantees to end users.

b. the NP applies a more liberal policy accepting a higher degree of multiplexing gain and consequently the network resource utilization will be higher but offering weaker guarantees to end users.

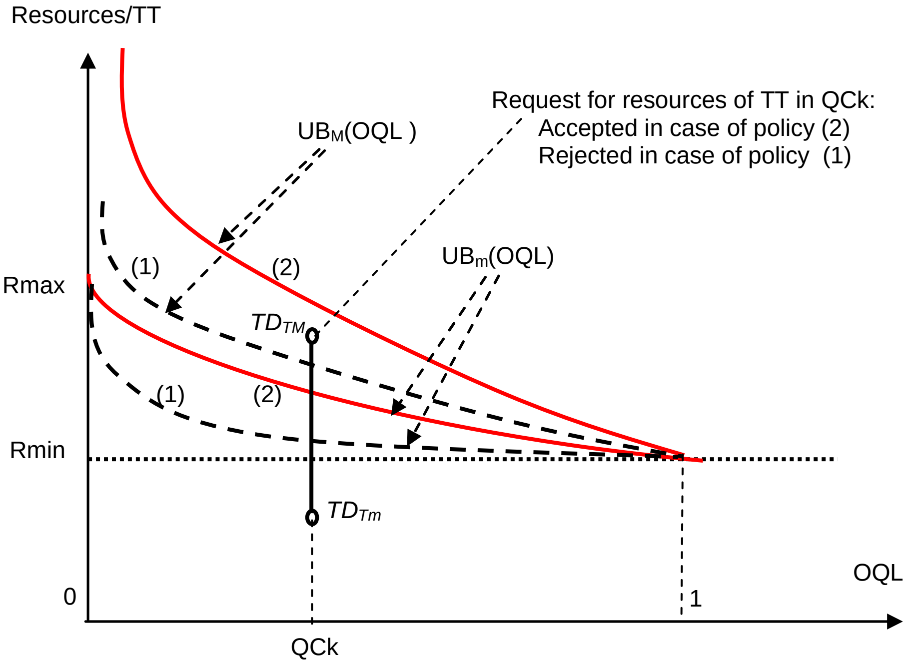

Figure 5: Applying different policies for pSLS Admission Control

Additionally the switch between different curves giving the upper bounds can be influenced by the congestion related measures given by the monitoring system at the network level.

6. Conclusions

An admission control algorithm, model based, is proposed to establish aggregated QoS controlled pipes in IP mult-domain environment. The implementation is simple but no evaluation of QoS degradation is possible apriori. Open issues leading to enhancement of the solution can be : studying the appropriate forms of UB_M and UB_m functions, in order to optimise the performance; combine the model based AC decisions with measurement information delivered by a monitoring system; comparative study of different methods to aggregate the new requests to the existing estimated traffic demands; study of how functions UB_M and UB_m functions can be selected by policies out of families of such functions.

References

[1] M.Grossglauser and D.Tse “A Framework for Robust Measurement-Based Admission Control”, IEEE/ACM Transactions on Networking, June 1999, Vol. 7, No. 3, pp.293-309.

[2], L.Breslau, E.Knightly, S.Shenker, I.Stoica, and Z.Zhang “Endpoint Admission Control: Architectural Issues and Performance”, SIGCOMM 2000, Stockholm 2000.

[3] P.Trimintzios, I.Andrikopoulos, G.Pavlou, P.Flegkas, D. Griffin, P.Georgatsos, D.Goderis, Y.T’Joens, L.Georgiadis, C.Jacquenet, R.Egan, “A Management and Control Architecture for Providing IP Differentiated Services in MPLS-Based Networks”, IEEE Communcations Magazine, May 2001, pp. 80-88.

[4] T.Damilatis, et.al., “Final Architecture, Protocol and Algorithm Specification”, TEQUILA IST Project Public Deliverable, D3.4 Part B, Oct. 2002.

[5] E.Mykoniati, C.Charalampous, P.Georgatsos and T.Damilatis, D.Goderis, P.Trimintzios and G. Pavlou, D.Griffin, Admission Control for Providing QoS in DiffServ IP Networks: The TEQUILA Approach, IEEE Communications Magazine January 2003, pp38-44.

[6] S. Georgoulas, P. Trimintzios, and G. Pavlou, "Joint Measurement- and Traffic Descriptor-based Admission Control at Real-Time Traffic Aggregation Points", Proc. IEEE Int. Conference on Communications (ICC2004), QoS and Performance Symposium, Paris, France, June 2004.

[7] P. Morand, et. al.,"Final specification of protocols and algorithms, for inter-domain SLS management and traffic, engineering for QoS-based IP service delivery" MESCAL IST Project Public Deliverable, D1.3, July 2005, www.mescal.org

[8] H. Asgari, ed., et.al., "Specification of protocols, algorithm, and components, the architecture, and design of SLS Management", ENTHRONE IST Project Public Deliverable D24F, July 2005, http://www.enthrone.org

[9] T.Ahmed - ed. Et al., "End-to-end QoS Signalling & Policy-based Management Architectures", ENTHRONE IST Project Public Deliverable D23F, September 2005, http://www.enthrone.org.

[10] E.Borcoci , A.Asgari, N.Butler, T.Ahmed, A.Mehaoua, G.Kourmentzas, S.Eccles " Service Management for End-to-End QoS Multimedia Content Delivery in Heterogeneous Environment", AICT Conference, July 2005, Lisbon.

[11] P. Morand, M. Boucadair, T. Coadic, P. Levis, R. Egan, H. Asgari, D. Griffin, J. Griem, J. Spencer, M. Howarth, N. Wang, S. Georgoulas, K.H. Ho, P. Flegkas, P. Trimintzios, G. Pavlou, P. Georgatsos, E. Mykoniati, I. Liabotis, T. Damilatis, "Final specification of protocols and algorithms, for inter-domain SLS management and traffic, engineering for QoS-based IP service delivery" MESCAL Deliverable, D1.3, July 2005

[12] M.Grossglauser and D.Tse "A Framework for Robust Measurement-Based Admission Control", IEEE/ACM Transactions on Networking, June 1999, Vol. 7, No. 3, pp.293-309.

[13] L.Breslau, E.Knightly, S.Shenker, I.Stoica, and Z.Zhang "Endpoint Admission Control: Architectural Issues and Performance", SIGCOMM 2000, Stockholm 2000.

[14] G.Iannaccone, M.May, and C.Diot "Aggregate Traffic Performance with Active Queue Management and Drop from Tail", Computer Communications Review, July 2001.

[15] R.Guerin, H.Ahmadi, and M.Naghshieh "Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks", IEEE Journal on Selected Areas in Communications, September 1991, Vol. 9, No. 7, pp.968-981.

[16] D.Eun and N.Shroff "A Measurement-Analytic Approach for Qos Estimation in a Network Based on the Dominant Time Scale", IEEE/ACM Transactions on Networking, April 2003, Vol. 11, No. 2, pp. 222-235.

[17] S.Belenki "An Enforced Inter-Admission Delay Performance-Driven Connection Admission Control Algorithm", ACM SIGCOMM, April 2002, Vol. 32, No. 2, pp. 31-41.

[18] C.Chuah, L.Subramarian, and R.Katz "Furies: A Scalable Framework for Traffic Policing and Admission Control", U.C Berkeley Technical Report No. UCB/CSD-01-1144, May 2001.

[19] T.Lee and M.Zukerman "Admission Control for Bursty Multimedia, Traffic", Proceedings of IEEE INFOCOM 2001, pp.478-487.

[20] S. Georgoulas, P. Trimintzios, G. Pavlou and K. H. Ho, "Measurement Based Admission Control for Real-time Traffic in IP Differentiated Services Networks," Proc. ICT 2005, Cape Town, South Africa, 3-6 May 2005.

[21] S. Georgoulas, P. Trimintzios, G. Pavlou, "Admission Control Placement in Differentiated Services Networks," Proc.IEEE Symposium on Computers and Communications (ISCC'2004), Alexandria, Egypt, IEEE, June 2004.

[22] S. Georgoulas, P. Trimintzios, and G. Pavlou, "Joint Measurement- and Traffic Descriptor-based Admission Control at Real-Time Traffic Aggregation Points", Proc. IEEE Int. Conference on Communications (ICC2004), QoS and Performance Symposium, Paris, France, June 2004.