ELSEVIER
Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 51 (2007) 4669–4678
Computer Networks
www.elsevier.com/locate/comnet

Service invocation admission control algorithm for multi-domain IP environments

Eugen Borcoci [a,*], Georgios Kormentzas [b], Abolghasem (Hamid) Asgari [c], Toufik Ahmed [d]

[a] *Universitatea Politehnica Bucuresti (UPB), Romania*
[b] *NCSR "Demokritos", Greece*
[c] *Thales Research and Technology (TRT), UK*
[d] *CNRS-LaBRI Laboratory, University of Bordeaux, France*

Abstract

One approach to solve the end to end (E2E) quality of services (QoS) problem of multimedia services delivery over multi-domain IP heterogeneous network infrastructures in a scaleable manner is the establishment of long term QoS enabled aggregated pipes. To allow dynamism, the actual service invocation of these pipes can be made as a separate action from the pipes subscription. This paper proposes a service invocation admission control algorithm that can be applied to aggregated IP pipes taking into account new service requests and the actual utilization of the domain resources of the subscribed pipes.

Keywords: Admission control; Heterogeneous networks; Invocation; Service management; SLA/SLS; Subscription

---

1. Introduction

In an end-to-end audio-visual chain of a next generation network is going to involve several entities such as *Service Providers* (*SP*), *Content Providers* (*CP*), *Network Providers* (*NP*), *Content Consumers* (*CC*), *Access Providers* (*AP*), *Brokers/Resellers*, etc. The transport of multimedia content from CPs' content servers (CS), through several heterogeneous IP autonomous domains, to potential CCs at a desired level of QoS raises a significant scalability problem. The establishment of logical long-term QoS-enabled pipes at an aggregation level over underlying heterogeneous IP multi-domains could constitute a scalable mechanism towards an end-to-end multimedia content delivery with QoS guarantees [1–5]. The pipes are logically established through negotiation of *Service Level Agreement/Specification* contracts between SPs (*pSLA/pSLS*).

The pipe construction is initiated by a SP, based on its knowledge about location of CPs/CSs and location of potential customers. SP makes a request for a pipe to involved NPs. The relationship between SP and NPs could be a star, a hub or a

---

[*] Corresponding author.
E-mail addresses: Eugen.Borcoci@elcom.pub.ro (E. Borcoci), gkorm@iit.demokritos.gr (G. Kormentzas), Hamid.Asgari@thalesgroup.com (A.(H.) Asgari), tad@labri.fr (T. Ahmed).

1389-1286/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2007.06.012

cascaded peering one [3,4]. The latter is more scalable because the SP does not need to interact with all NPs in the chain, but only with the first one situated at the ingress of the desired path. Each pSLS request contains all desired QoS parameters (e.g., bandwidth, delay, jitter, loss rate, etc.). The pipes are requested by SPs and they are agreed between SP/NP, NP/NP, etc. This is called the *subscription* phase. The actual network resources allocation at the underlying network elements for these pipes can be done immediately at the time of subscription, or later, based on agreed explicit or implicit requests signaled by the SP. This action is called aggregate pipe *service invocation*.

Note that we make a clear distinction between subscription and invocation of the aggregated pipes and individual user service subscription and/or invocation. After the installation of the aggregated pipes in the network domains, SP is able to offer services for individual flows. The aggregated capacities are "sold" in a retail manner, to many customers, through individual contracts *customer-SLA/SLS (cSLA/cSLS)* between SP and each interested customer.

Focusing on the aggregated pipe invocation phase, this phase is the one in which the actual QoS enabled aggregated pipe is installed at SP request in the network elements of each involved NP in the pipe chain. The amount of requested SP resources may be those previously agreed in *pSLS* contracts, or may have different values (less or even more if over-subscription is allowed). Therefore an invocation-level Admission Control (AC) is necessary in each domain, taking into account the service requests and the actual utilization of the underlying domain resources.

This paper proposes such an AC algorithm for *pSLS* based pipes invocation. Starting from previous other approaches, [1,2,6,7,10], it is proposed and studied a modified AC, flexible and scalable, policy driven and having a simple implementation.

The paper is organized as follows. Section 2 briefly discusses the pSLS invocation framework where the proposed AC algorithm could be applied. Section 3 presents the proposed service invocation admission algorithm and Section 4 concludes the paper.

2. pSLS invocation framework

Fig. 1 depicts a high level view of Traffic Trunks (TTs). Each TT belongs to a QoS class of service (QC). TTs are considered distinctly for intra and inter-domain paths/links. An intra-domain TT can be defined from an input I/F of an ingress router up to an output I/F of an egress router. An inter-domain TT can be defined from an output I/F of an egress router up to an input I/F of an ingress router of the next domain.

The intra-domain TT constitutes an abstraction of resources allocated for this TT between an ingress and egress point of this domain, no matter the intra-domain path is. On an inter-domain link, we suppose that we have one TT per QC.

The pSLS invocation framework that is deploying in the context of the IST FP6-IP ENTHRONE project [3,4] is depicted in Fig. 2. The *pSLS Invocation Handler* (*pSLS_IH*) has the role to activate the *pSLSs*, at request of SP, after the *pSLSs* have been agreed and subscribed.

To process a new pSLS invocation request, an Admission Control (AC) algorithm has to consider the following information: *subscription information* (read from the pSLS repository) for the QoS class in question; *invocation request parameters* for an already agreed pSLS pipe belonging to a certain QoS class; *previous invocation parameters* of the

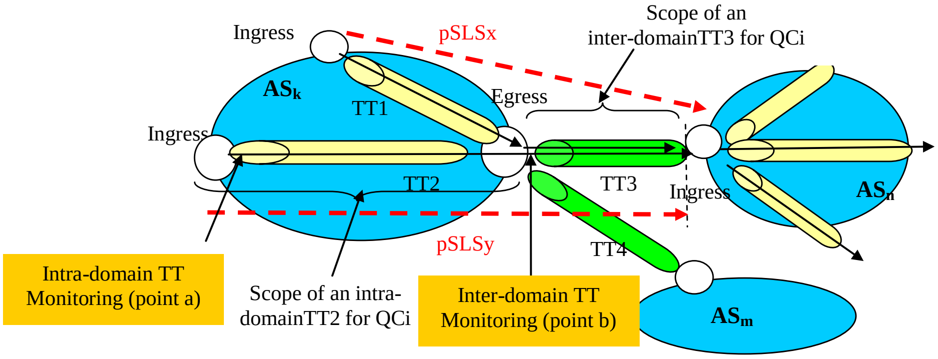

Fig. 1. Scopes of intra and inter-domain traffic trunks.

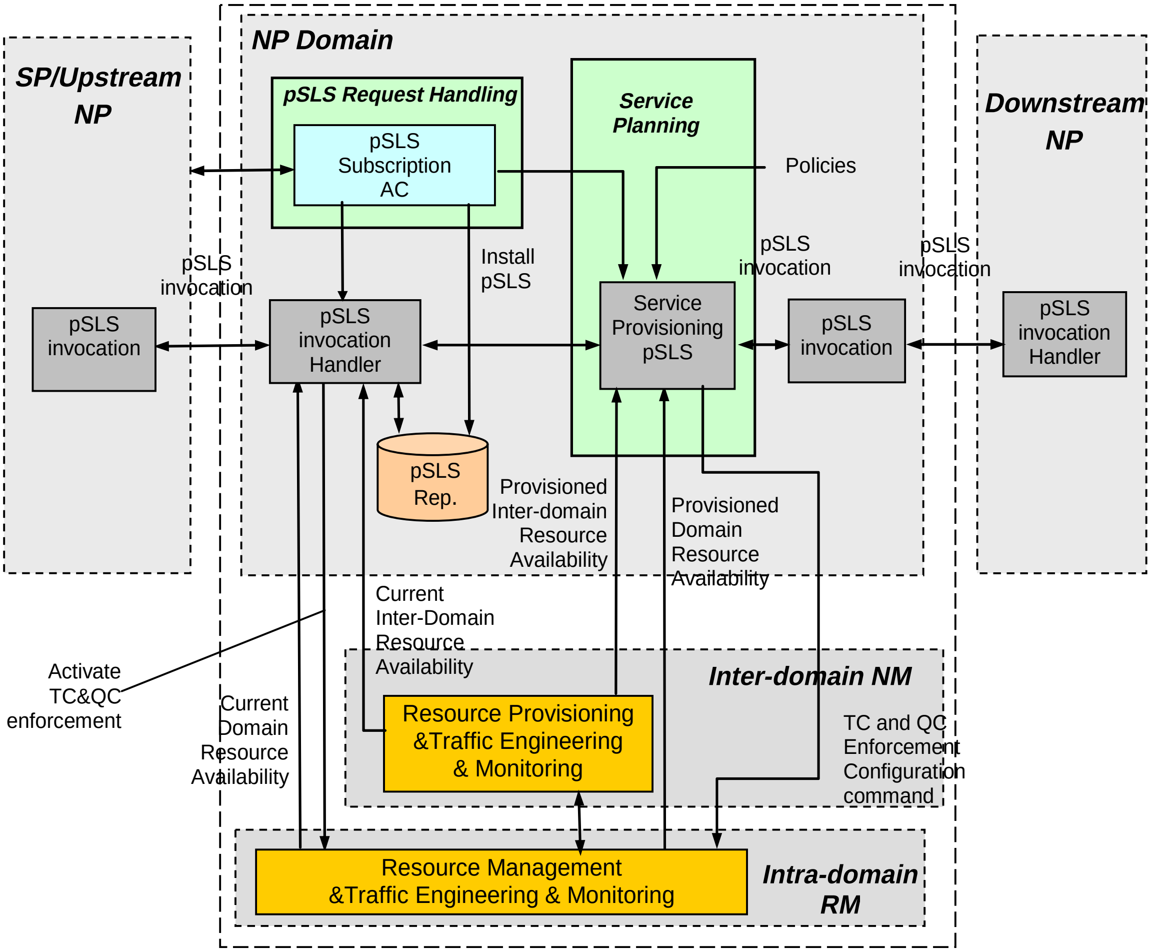

Fig. 2. pSLS Invocation framework functional blocks.

request – if some previous invocation exists for the same pipe, but with different parameters; *current status of the network load* (with respect of the path desired), delivered by the network monitoring system; *policy rules*, which can influence the admission or rejection of the invocation request. In case of pSLS invocation request acceptance, the configuration information for network level AC and traffic conditioning purposes are downloaded to the NP's intra-domain Resource Manager.

The *pSLS_IH* has the following interfaces with the respective operation:

1. *pSLS Subscription AC* (*pSLS-S-AC*) *to pSLS_IH*: Information related to new pSLSs agreed by pSLS-S-AC is passed to the handler, or, this can be obtained by pSLS_IH from pSLS repository.
2. *SP/Upstream NP pSLS Invocation to pSLS_IH*: It supports the explicit invocation start or termination of a pSLS by a SP/Upstream NP. The handler notifies the requesting party about success or failure of its request.
3. *pSLS_IH to Service Provisioning*: It allows for prolongation of pSLS invocation to downward domains via *Service Provisioning* and also allows the *pSLS_IH* to interrogate the *Service Provisioning* about the validity of pSLS invocation request.
4. *Intra-domain Resource Manager* (*RM*) *to pSLS_IH*:
   - Provisioned resource availability for the domain – *Domain Total Trunk Resource Availability Matrix (DTT_RAM)* described in [3,4]. This matrix describes what traffic trunks (scope, characteristics) have been provisioned inside the domain for each QC. Alternatively, this information can be obtained from the *Service Provisioning* block, which also has received this matrix from the NP.

   - *Current Domain Resource Load Matrix*-(*CD-RLM*), this matrix should be offered by the *Network Monitoring System* (after measurements performed on the network). It shows the fraction of busy resources out of the total capacity of each TT of *DTT_RAM*.
5. *Inter-domain Network Manager* (*NM*) *to pSLS_IH*:
   - Provisioned resource availability for the output links of the domain – *Inter-domain Total Trunk Resource Availability Matrix* (*ITT_RAM*) described in [3,4]. It contains data on the output traffic trunks for each QC. The traffic trunks are external but adjacent to this domain. Alternatively this information can be obtained from the *Service Provisioning* block, which also has received this matrix from the NP.
   - *Current Inter-domain Resource Load Matrix* – (*CI-RLM*) – given by the *Network Monitoring System*; it indicates the amount of busy resources out of the total specified by *ITT_RAM*, for each TT.
6. *pSLS_IH* to *Intra-domain Resource Manager* (*RM*): This interface allows the actual activation of traffic conditioning and QC enforcement at the involved network elements.
7. *pSLS Invocation Handler* to *pSLS repository:* Information about subscribed pSLSs is passed to *pSLS_IH*. Information about previous invocation parameters are also stored in the *pSLS repository* and can be read by the *pSLS_IH*. The *pSLS_IH* writes or updates in the *pSLS repository*, the invocation parameters for each invoked pSLS.

Optionally the *pSLS Invocation Handler* can inform the *Intra-domain RM* about special situations, e.g., “a large number of rejections of pSLS invocation requests” in order to trigger network reconfiguration.

3. The pSLS invocation admission control algorithm

3.1. Requirements and assumptions

A pSLS may be invoked in several ways, depending on the scheduling agreements existent in the pSLS contract:

(a) *Static invocation*: this is done immediately after subscription, (no additional signaling from SP); each NP on the path, installs the pipe in its network domain, using the full range of parameters (bandwidth) as specified by pSLS contract. The lifetime of the pipe installed is the same as specified in the pSLS contract.

(b) *Dynamic explicit invocation*: this is done at SP initiative at an invocation request instant, (it may be later than subscription), for the full bandwidth previously agreed, or only a percentage of it. The *pSLS_IH* at the first NP receives invocation information, and extracts the relevant parameters. The Service Provisioning block at NP will contact the next NP along the path of the pSLS associated pipe in order to extend the invocation chain of actions, and so on, up to the last domain of the pipe. The termination of invocation is also explicitly signaled by SP.

(c) *Dynamic implicit invocation*: if this was agreed in the pSLS contract (time schedule with desired time intervals for invocation) then each NP on the path, will generate the invocation commands and latter termination.

This paper analyses the case *b*, which is the most complex one. Note that the pSLS subscription process (depending on domain policy) can accept over-subscription, based on some multiplexing gain assumptions/policies when admitting the pSLS requests. Therefore the *pSLS-IH* should include an AC algorithm to analyse the actual parameters desired in the invocation request.

The proposed pSLS-I-AC (pSLS invocation admission control) algorithm complements the pSLS-S-AC (pSLS subscription admission control) algorithm of Fig. 2. The main difference between the *pSLS-S-AC* and the proposed *dynamic pSLS-I-AC* algorithm is that the latter has to take also into account the real network load in addition to the pSLS contract parameters. Furthermore, at the invocation time it has to satisfy the new user performance QoS requirements, to avoid jeopardizing the existing admitted flows and to allow for efficient network resources utilization.

The proposed pSLS-I-AC has also to comply with the following requirements:

- work at the aggregated level, for traffic trunks associated for different classes of services (aggregated pipes associated to pSLS contracts),

- apply AC decisions appropriate for *real time traffic flows*,
- allow dynamic, total or partial invocation of previously subscribed pSLSes, while taking into account the current status of the network load,
- base its decisions on information about:
  - *pSLSs* established, taken from the *pSLS Repository* as a result of previous *pSLS* accepted subscriptions,
  - current level of aggregated traffic load, (already invoked *pSLSs*); this information is obtained from an appropriate monitoring system,
  - available network resources (upper bounds),
  - traffic description for new *pSLS* invocation requests,
  - information based on the enforced policies for each domain.

In a first, more simple approach, we do not consider the potential interactions among different QCs, despite the fact that such an effect can exist in practice. It is supposed that resource provisioning by the network is done per QoS class; TTs concerning different QCs seen as separate resources, irrespective the fact that on the physical infrastructure the paths can be (totally or partially) the same. At the network level, the traffic conditioning mechanisms and scheduling are supposed to differentiate between classes and apply the appropriate scheduling, so as to fulfill the requirements of each QC. In a more complex approach, the above restriction can be relaxed (resource sharing among several TTs belonging to different QCs).

3.2. Total traffic demand estimation

We adopt the approach described in [2,7] for a hybrid traffic-descriptor and measurement-based AC, where the new flow to be admitted is described by its own traffic descriptor, while the bandwidth consumed by the already established flows is obtained by real-time measurements through an appropriate monitoring task system. Taking decisions based on the actual network status, makes less important the accuracy of the traffic descriptors and the conformance of the real flows to those descriptors.

- Traffic multiplexing issues

The pSLS associated pipe carries many multiplexed individual flows. Therefore an *effective bandwidth based* approach is appropriate for bandwidth related computations and allocation. In [8] it is shown that when the statistical multiplexing is significant, the distribution of the stationary bit rate can be accurately approximated by a Gaussian distribution. The work [9] shows even more: aggregation of a fairly small number of traffic streams is still sufficient to allow for the Gaussian characterization of the input process. The effective bandwidth of the multiplexed sources is given by [7]:

$$C \approx m + a\sigma, \quad \text{where} \quad a = [-2ln(err) - \ln(2\pi)]^{1/2}. \tag{1}$$

Here, m is the mean value of the *aggregated* bit rate; σ is the *standard deviation* of the aggregate bit rate; *err* is the upper bound on allowed queue overflow probability.

- Measuring issues

Fig. 3 depicts an example of a domain having several trunks on the same physical link. R1 is an ingress router and R2 an interior router of a domain (AS). TT1 has three active pipes (i.e., already invoked) corresponding to three *pSLSs*. A fourth invocation request arrives for a *pSLS* pipe in TT1.

The monitoring system is supposed to be able to measure the traffic load *per each TT*. This assumption requires the capability *to measure load on each of the different TT flow belonging to the same QC*. For intra-domain TTs, the measuring process is performed at the output I/Fs of each ingress router of the domain (e.g., R1), point a (see both Figs. 1 and 3). For inter-domain TTs, this is done at the output interfaces of each egress router of the domain, point b (see Fig. 1).

Considering an intra-domain TT as an edge-to-edge logical pipe and an inter-domain TT as a domain-to-domain logical pipe, the load information measured in the two aggregation points a. and b. is sufficient to know the load on each TT. This can lead to a scalable solution for monitoring. Assuming the measuring can be done, at the output interfaces in cases a. or b., *the measured* (for a given TT) parameters are the mean rate M_{ms} of the offered load and its variance $(\sigma_{\text{ms}})^2$. These values are in fact measured at the input of intra-domain TTs and respectively at the input of inter-domain TTs.

The paper does not discuss the selection of the time window value for measurements. Details are

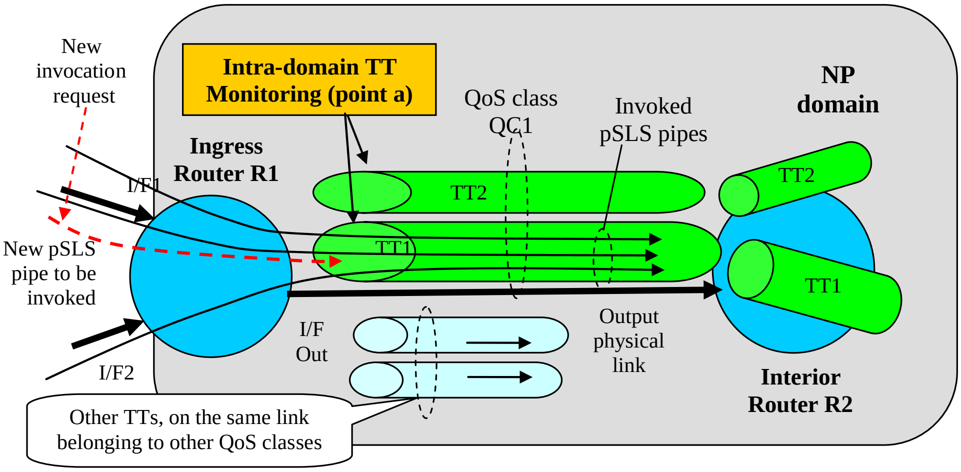

Fig. 3. pSLS Invocation scenario-example.

given in [2,7,10], which propose a sufficient time window to take measurement samples.

- Combining the requested traffic descriptor with the measured values

We suppose that a new invocation request is coming to the proposed pSLS-I-AC algorithm, invoking partially or totally a pSLS pipe transported by a traffic trunk (TT), belonging to a given QoS traffic class QC1. TT has some resource bounds *UB_m(OQL1)-minimum* and *UB_M(OQL1)-maximum*, where the factor *Overall Quality Level (OQL)* can be found [3,4]. Let OQL1 be the value of QC1. The overall quality level is a factor (OQL $\in [0,1]$) selected by the NP for serving different QCs. The greater OQL, the better QoS guarantees are obtained meaning less possibility of sharing resources between TTs and thus less degree of overbooking admission for requests. The *pSLS-I-AC* algorithm compares the sum of the current load (measured) and the new requested load against the above bounds.

The new request comes with the parameters (m_{new}, σ_{new} – they should be compliant with the values of the pSLS contract). Now, we have to "add" this traffic to the measured one in order to estimate the total invocation demand (including the new requested flow). This will give the necessary invocation capacity estimation:

$$C_{i-est} = (M_{ms} + m_{new}) + a'_{PLR}[(\sigma_{ms})^2 + (\sigma_{new})^2]^{1/2}, \tag{2}$$

where a'_{PLR} is computed as in (1) for the target *packet loss rate* (PLR) for the aggregated flow.

In the particular case when the new request specifies only its peak rate p_{new-r}, then $\sigma_{new=0}$ and $m_{new} = p_{new-r}$.

Having this sum, the admission control criterion is very simple: *If* ($C_{i-est} \leqslant PT^* C_{avail}$) *then accept, otherwise reject.*

Here C_{avail} is the available capacity (upper bound). We also introduce a *Protection Threshold* ($PT \in [0,1]$) factor, which is the percentage of the resources allowed to be used. This factor might compensate for some measurement errors, but also may be influenced by NP policies. For instance, if the NP network manager would like to be more conservative w.r.t. resource allocation, then the PT is lower, or if it would like to be more liberal, then the PT is greater.

- Refinements for PT computation

Works [2,7] propose a different factor, called *precaution factor (PF)* $\geqslant 1$, and a heuristic formula for *PF*. This factor is included in the AC criterion, in order to address such issues as: (1) source heterogeneity and (2) effect of measurement errors. The decision rule in their case is:

$$\textbf{If } PF^* C_{i-est} \leqslant C_{avail} \quad \textbf{then} \text{ accept, otherwise reject.} \tag{3}$$

A second correction factor is proposed in [2,7], that is $PF' = PF^* c$, where c depends on the PLR value for the reference source and target PLR for aggregated flow.

We propose to define:

$$PT = \alpha(1/PF), \quad \text{with} \quad \alpha \in (0,1], \tag{4}$$

where the factor α can be determined by policy considerations. The introduction of α enables the control of the resource allocation process by a single tuning factor at a high level.

- Specific decision formulas

We use the generic notation $C_{\text{available}}$ to indicate the $UB_{\mathrm{m}}(OQL1)$ bound or $UB_{\mathrm{M}}(OQL)$. Remember that the two bounds correspond and have to be compared to the minimum and maximum requested rates:

$\boldsymbol{SR}_{\mathrm{m}}$	minimum traffic rate to be offered to an SLS so that its service can run at acceptable level.
$\boldsymbol{SR}_{\mathrm{M}}$	maximum traffic rate to be offered to an SLS so that its service can run at very good level.

If we use a reference source model as defined in this section then we can roughly approximate the minimum rate requested as $SR_{\mathrm{m}} = m$, and $SR_{\mathrm{M}} = m + a\sigma$; see formula (1). Therefore the comparison relationships used in decision, $C_{i-est} \leqslant PT^{*}C_{\text{available}}$, applied for a QoS class QC1, on a traffic trunk TT1, having a OQL factor OQL1 would become:

$$TD_{\mathrm{Tm}} = (M_{\mathrm{ms}} + m_{\mathrm{new}}) \leqslant PT^{*}UB_{\mathrm{m}}(OQL1) \tag{5}$$

for the lower bound, and

$$TD_{\mathrm{TM}} = (M_{\mathrm{ms}} + m_{\mathrm{new}}) + a'_{\mathrm{PLR}}[(\sigma_{\mathrm{ms}})^2 + (\sigma_{\mathrm{new}})^2]^{1/2} \leqslant PT^{*}UB_{\mathrm{M}}(OQL1) \tag{6}$$

for the upper bound.

In order to increase the ability to react to traffic variation and network load and prevent QoS degradation, we extend (additionally to the above PT factor), the usage of formulas (5) and (6) with a *Random Early Detection (RED)*-like behaviour for AC decision. A similar approach is used in [11], but in a different context.

3.3. Algorithm description

The pSLS subscription phase establishes two kinds of commitments for a pSLS pipe belonging to a given TT through the pSLS-S-AC algorithm:

- strong commitments if the total resource requests is below the *Rmin*,
- statistical commitment if the total resource requested is between *Rmin* and *Rmax*.

The above commitments could be found in detail in [3,4]. The upper bounds $UB_{\mathrm{m}}(OQL)$ and $UB_{\mathrm{M}}(OQL)$ values are defined for each QC. The portion of resources between *Rmin* and *Rmax* is a shared pool of resources between this TT and other TTs (see Fig. 4). Therefore it is a matter of NP policy how many requests will be admitted, if the total amount of resource request for a given TT is close to the value *Rmax*. The proposed *pSLS-I-AC* algorithm should prevent entering in to the congestion region for the trunks.

The pSLS invocation parameters could be different from the pSLS subscription parameters. If the pSLS contract specifies permission of overbooking then the system can allow a larger amount of requests in the invocation request than in the subscription contract. But, this should be controllable in the sense that congestion should be avoided.

- Approach 1 (monitoring system can supervise each TT)

We suppose that we have knowledge on:

- a TT_i, belonging to the QoS class QC_j;
- the values *Rmin, Rmax* for TT_i and the bounds $UB_{\mathrm{m}}(OQL_j)$ and $UB_{\mathrm{M}}(OQL_j)$ (remember that the portion *Rmax-Rmin* is shared with other TTs, belonging to the same QC);
- pSLS_k subscription information for a pipe on TT_i (this is not yet invoked);
- the measured values of the total load, at the output interfaceof the ingress router invoked on TT_i.

Under such conditions a new invocation request comes for invocation of pSLS_k.

The high level description of the algorithm is:

- verify that the request is conformant with the subscription and reject if not. (Note that if overbooking is allowed then the request can ask for more resources than in the previous subscription.)

If verification is ok then,

- compute the total request current load plus the new request (minimum and maximum),

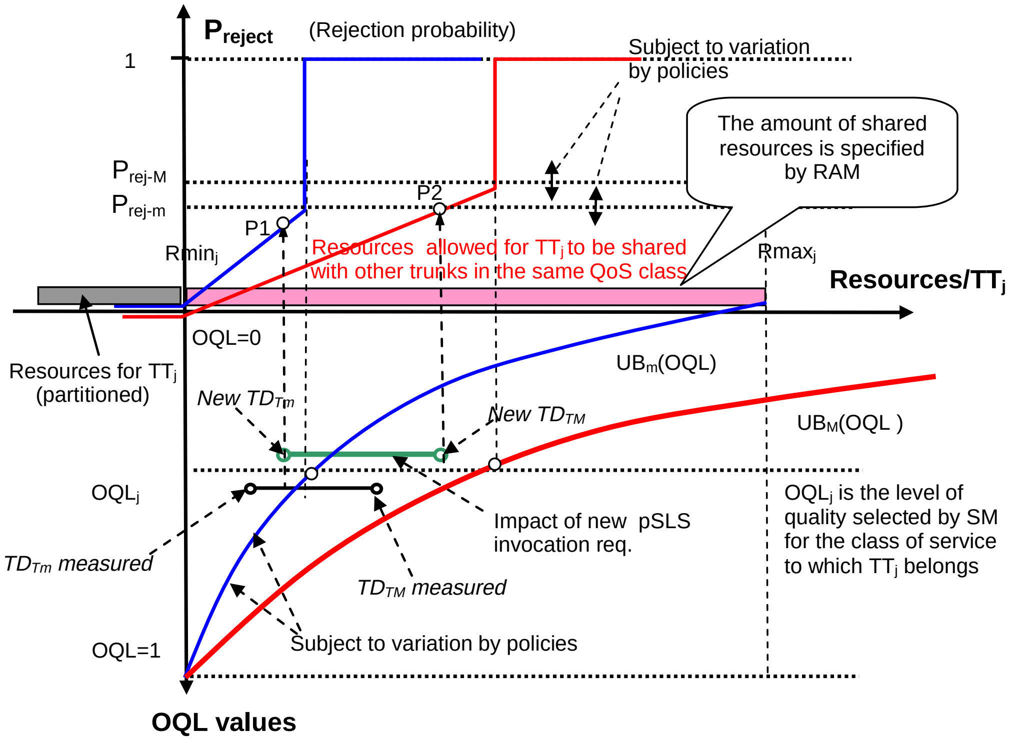

Fig. 4. pSLS-I-AC algorithm (traffic-descriptor based + measurement based + RED) P1, P2 are rejection probabilities for min and respectively max demand values.

– if the request-maximum level is below the *Rmin* level of resources then accept it, otherwise apply the RED policy as described in the Fig. 4. Therefore, the decision to accept/reject a request, which brings the total demand in the region *Rmin*–*Rmax* will be not a binary one as in formulas (5) and (6) but a probabilistic one. In Fig. 4, for simplicity reasons we have put $PT = 1$.

Note that on different links along the path of a given trunk TT_j inside a domain, this trunk may share bandwidth with other TTs, (belonging to the same QoS class) but the latter ones may be different on each link. So, for scalability reasons, the algorithm is *not aware* of the amount of the current load produced by other TTs sharing bandwidth with TT_j. This weakness can be compensated by the RED mechanism. While total load on each TT is closer to its *Rmax* value, the more probable is that the RED mechanism will reject a new invocation (see Fig. 4). So the AC decision for a pSLS invocation request addressed to a given TT, is decoupled from invocations belonging to other TTs. But the actual coupling (from the performance point of view) still exists between different TTs if the shared part of the resources is used. Concerning this point, one has to adjust the RED parameters and $UB_{\rm m}$ and $UB_{\rm M}$ bounds through policy based actions, taking into account information delivered by the monitoring system.

- Approach 2 (monitoring system cannot supervise each TT)

In this case the monitoring system is not able to measure individual TT load, for each TT belonging to a given QC, but only aggregated values for traffic load for different QCs. The formulas (5) and (6) cannot be used. In such a case, to compute the current load on the trunk in question, the algorithm will *use the invocation values for the active pSLSs* and add to these the new request's desired load (specified by traffic descriptors). The RED mechanism described for Approach 1 can be still used. Additionally one can use the measured values given by the monitoring system in order to adjust the *Protection Threshold* (PT) proposed in the formula (4) or to adjust the factor α via an appropriate policy.

For scalability reasons, we constrain *the resource sharing only among TTs having the same edges, the* *same intra-domain path and belonging to the same QoS class.* However, the network utilization could be better if we allow sharing resources among TTs belonging to different QoS classes. This will need an extension of the algorithm, which is left for further study.

4. Conclusions

An admission control algorithm has been proposed for invocating the installation of QoS enabled aggregated pipes in a multi-domain environment. It is believed that such an algorithm could enhance the dynamism of an E2E QoS-offering system, by decoupling the subscription phase for QoS enabled pipes and their actual invocation and installation in the network. The proposed AC is flexible in resource allocation, due to its decision method, in order to get different levels of QoS guarantees. It could be easy to implement, provided that exists a capable network monitoring system in each autonomous domain. Also it can be extended to be driven by a policy-based management logical structure.

References

[1] E. Mykoniati, C. Charalampous, P. Georgatsos, T. Damilatis, D. Goderis, P. Trimintzios, G. Pavlou, D. Griffin, Admission control for providing QoS in DiffServ IP networks: the TEQUILA approach, IEEE Communications Magazine (January) (2003) 38–44.

[2] 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, www.mescal.org.

[3] H. Asgari, et al. (Eds.), 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.

[4] T. Ahmed, et al. (Eds.), End-to-end QoS signalling and policy-based management architectures, ENTHRONE IST Project Public Deliverable D23F, September 2005, http://www.enthrone.org.

[5] E. Borcoci, A. Asgari, N. Butler, T. Ahmed, A. Mehaoua, G. Kormentzas, S. Eccles, Service management for end-to-end QoS multimedia content delivery in heterogeneous environment, in: Proceedings of AICT Conference, July 2005, Lisbon.

[6] M. Grossglauser, D. Tse, A Framework for Robust Measurement-Based Admission Control, IEEE/ACM Transactions on Networking 7 (3) (1999) 293–309, June.

[7] S. Georgoulas, P. Trimintzios, G. Pavlou, Joint Measurement- and Traffic Descriptor-based Admission Control at Real-Time Traffic Aggregation Points, in: Proceedings of the IEEE International Conference on Communications (ICC2004), QoS and Performance Symposium, Paris, France, June 2004.

[8] R. Guerin, H. Ahmadi, M. Naghshieh, Equivalent capacity and its application to bandwidth allocation in high-speed networks, IEEE Journal on Selected Areas in Communications 9 (7) (1991) 968–981, September.

[9] D. Eun, N. Shroff, A measurement-analytic approach for QoS estimation in a network based on the dominant time scale, IEEE/ACM Transactions on Networking 11 (2) (2003) 222–235. April.

[10] S. Belenki, An enforced inter-admission delay performance-driven connection admission control algorithm, ACM SIGCOMM, April 32(2) (2002) 31–41.

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

Eugen Borcoci is professor at University "Politehnica" of Bucharest (UPB), Electronics, Telecommunication and Information Technology Faculty. He graduated in Electronics and Telecommunications at Polytechnic Institute of Bucharest and then in Mathematics at University of Bucharest. He obtained his Ph.D. degree at Polytechnic Institute of Bucharest. His current interest is in telecommunication/computer systems and networks: architectures for fixed and mobile communications, signaling and protocols, QoS issues, multicast and multimedia communications. He has been UPB team leader in several European IST research projects: COP #62 Project (1995–1997); KIT project No. 977124 (1998–2000); IST-200-25137-MOICANE – (2001–2002), Network of Excellence EuroNGI IST 507613 (2003–2006), (www.eurongi.org), ENTHRONE Project IST 507637, (2003–2005), WEIRD (2006–2008) (www.ist-weird.eu). He has been member of several Int'l Conferences Technical Committees: PSTV'95, CFIP'96, 97, 99, IEEE ICT 2001, 2002, Eurasia ICT 2002, FORTE 2003, MoMM 2004, NGI 2005, AICT 2005, 2006, NGI 2006, NGI 2007. He published 5 books, 4 textbooks and over 140, scientific or technical papers and research reports. He participated to more than 30 scientific or technical national projects.

Georgios Kormentzas is currently Assistant Professor in the University of the Aegean, Department of Information and Communication Systems Engineering. He received the Diploma in Electrical and Computer Engineering and the Ph.D. in Computer Engineering both from the National Technical University of Athens (NTUA), Greece, in 1995 and 2000, respectively. He has been actively working for many years on the area of network management and quality of service of computer and communication systems where he has introduced the concept of abstract information model, an ancestor of next generation networking middleware, which is his main current research interest. He is a member of pronounced professional societies, an active reviewer for several journals and conferences and EU-Independent Expert for Marie Curie Actions. Currently, George Kormentzas is Technical Manager of IST-2005-FP6 UNITE STREP project: "Virtual Distributed Testbed for Optimization and Coexistence of Heterogeneous Systems" and co-chair of Beyond 3G System Architecture Cluster concerning FP6-IST projects.

Abolghasem (Hamid) Asgari (Hamid.Asgari@thalesgroup.com) received his B.Sc. from Dr. Beheshti University in Tehran, M.Sc. (Honors) from the University of Auckland, New Zealand both in Computer Science, and his Ph.D. in the design and analysis of ATM networks from University of Wales, Swansea in 1997. He was with Iran Telecom. Research Center (ITRC) from 1986 to 1990. He joined Thales (formerly Racal) Research & Technology (UK) in 1996 where he is an Chief Engineer/Project Manager specialising in the data communication systems/networks. He was involved in simulation, design and analysis of integrated IP services and network architectures. He has worked on a number of European Commission projects including IST TEQUILA, MESCAL and ENTHRONE projects. His main research interests are in network QoS management, QoS-aware monitoring, network and system level performance evaluation, traffic engineering, and service management.

Toufik Ahmed received his Master's and Ph.D. degrees in computer science from the University of Versailles in 2000 and 2003, respectively. He was a visiting scientist at the School of Computer Science of the University of Waterloo in 2002 and research fellow at PRiSM laboratory of the University of Versailles in 2003. He is currently an associate professor at ENSEIRB school of engineers/University of Bordeaux I. His main research activities concern Quality of Service for multimedia wired and wireless networks, end-to-end signalling protocols, and policy-based networking.