Finding Inter-domain QoS Enabled Routes Using an Overlay Topology Approach

Şerban Gerogică Obreja, *IEEE Member*
Communications Department, ETTI
University POLITEHNICA Bucharest
Bucharest, Romania
e-mail: serban@radio.pub.ro

Eugen Borcoci, *IEEE Member*
Communications Department, ETTI
University POLITEHNICA Bucharest
Bucharest, Romania
e-mail: eugen.borcoci@elcom.pub.ro

***Abstract*— The transport of multimedia flows over the Internet needs to manage and control end to end Quality of Services (QoS) at transport level. One problem, which needs to be solved in this case, is finding inter-domain QoS enabled paths. This paper deals with the problem of establishing QoS enabled aggregated multi-domain paths, to be later used for many individual streams. It is proposed a simple but extendable procedure, based on overlay approach, to find several potential inter-domain end to end paths. The problem of QoS path finding is spilled in two phases. First several paths between the source and destination domains are found using the overlay topology. Then, the process of QoS path finding is completed in a second phase, when QoS enabled aggregated pipes are established, by negotiation between the domains' managers, along one of the paths founded in the first phase. The subsystem proposed is part of an integrated management system, multi-domain, dedicated to end to end distribution of multimedia streams.**

Keywords- QoS routing, end to end QoS management, pSLS, overlay network topology.

I. INTRODUCTION

The real time multimedia services, delivered on Internet networks, raised new challenges for the network regarding the end to end (E2E) quality of services (QoS) control in order to ensure the proper delivery of the services from content provider (source) to content consumer (destination). But traffic processing in real Internet deployments is still mostly best effort. Several approaches have been proposed, focused on provisioning aspects – usually solved in the management plane - and then in the control plane: e.g., well known dynamic techniques have been standardized, like IntServ, Diffserv, or combinations. The routing or - more generally - QoS enabled path finding and then maintaining, are also a part of the scene. Offering multimedia services in multi-domain heterogeneous environments is an additionally challenge at network/ transport level. Service management is important here, for provisioning, offering, handling, and fulfilling variety of services. Appropriate means are needed to enable a large number of providers to co-operate in order to extend their QoS offerings over multiple domains. To this aim, an integrated management system can be a solution, preserving each domain independency but offering integration at a higher (overlay) layer in order to achieve E2E controllable behavior.

This paper deals with the problem of establishing QoS enabled aggregated multi-domain paths, to be later used for many individual streams. It is proposed a simple but extendable procedure, running at management level, to find (through communication between domain managers) several potential inter-domain end to end paths. Then, using a resource negotiation process performed also in the management plane, QoS enabled aggregated pipes are established. All these function are performed at an overlay level, based on abstract characterization of intra and inter-domain capabilities delivered by an intra-domain resource manager. The subsystem is part of an integrated management multi-domain system, dedicated to end to end distribution of multimedia streams.

The QoS path finding is not a traditional routing process: it is not implemented on routers, and it doesn't choose a route between network devices, but between two or more nodes of an overlay virtual topology described at inter-domain level. Together with the intra-domain QoS routing available inside each network domain we will obtain an E2E QoS routing solution.

The main advantage of this solution is that, by separating the process of path finding from the QoS negotiation, the path searching process doesn't need to work real time. So we can find several paths in very complex overlay topologies. Also, by simplifying the overlay topology, by considering only the domain managers as topology nodes, our solution will work for very complex topologies, being no need for a hierarchical approach.

This paper is organized as follows: the Section 2 contains the state of the art in QoS inter-domain routing; the Section 3 shortly describes the general Enthrone architecture focusing on the service management at the network level. The Section 4 introduces the proposed QoS inter-domain path finding solution. Section 5 presents details about the implementation and Section 6 contains conclusions and possibilities of extensions and open issues.

II. STATE OF THE ART

Because our approach deals with QoS path finding and routing, a short overview of the available approaches for QoS routing is presented below [13][14][15][17][18]. We distinguish between intra- and inter-domain QoS problems.

The intra-domain QoS routing solutions could be divided in two major approaches.

Classically, intra-domain QoS routing protocols run on the routers and find paths with QoS constraints from source to destination.

Other solutions are based on a domain central manager, having knowledge of the total resource allocation inside the domain, and use an algorithm to determine QoS routes between source and destination. In this case the QoS routing process is run by a dedicated module of the domain manager, and the resulted route is installed on the network equipments by a network controller. Usually the QoS routing process is triggered by a new request for a QoS path through the domain.

For inter-domain QoS routing also we can distinguish between two kinds of approaches. The first one proposes enhancements for the BGP protocol in order to support QoS features. The BGP advertises QoS related information between autonomous systems (ASes), and the routing table is build taking into consideration this additional QoS information. The Q-BGP protocol, proposed in MESCAL project [20], is such an example.

Another category of inter-domain QoS routing solutions are based on the overlay network idea [13][14]. An overlay network is built, which abstracts each domain with a node, represented by the domain service manager, or with several nodes represented by the egress routers from that domain. Then protocols are defined between nodes for exchanging QoS information, and based on this information QoS routing algorithms are used to choose the QoS capable path. In [13] a Virtual Topology solution is proposed. The VT is formed by a set of virtual links that map the current link state of the domain without showing internal details of the physical network topology. Then a Push and a Pull model for building the VT at each node are considered and analyzed. In Push model each AS advertise their VT to their neighbor ASes. This model is suited for small topologies. In Push model the VT is requested when needed, and only from the ASes situated along the path between source and destinations, path which is determined using BGP routing information. If BGP kept several routes between source and destination than the VTs for each domain situated along the founded paths are requested. Based on this VTs information the QoS route from source to destination is calculated. After that an end to end QoS negotiation protocol is used to negotiate the QoS resources along the path.

One problem with these solutions is that they are based on the virtual available resource topology information obtained from other ASes. This requirement could be not accepted by the actual network providers, due to their confidentiality policy regarding their resource availability.

Also, these solutions based on an end to end QoS negotiation process. After the QoS path is found, the negotiation process is started. The QoS routing process previously performed is increasing the chance of negotiation success, but it implies two QoS searching processes: building the QoS topology and secondly negotiation in order to reserve resources.

This paper proposes a simpler approach by separating the process of path searching from the process of QoS negotiation (QoS searching path). By combining these two processes we will obtain a QoS inter-domain routing solution.

This was developed and integrated in an E2E QoS management system [2][8][9][10]. The system was proposed and implemented by an European consortium in the FP6 European project ENTHRONE [2][3][4][5], and continued with ENTHRONE II [6][7][8]. The ENTHRONE project is an integrated management solution based on the end-to-end QoS over heterogeneous networks and terminals. It proposes an integrated management solution that covers the entire audio-visual service distribution chain, including protected content handling, distribution across networks and reception at user terminals.

The overlay QoS path finding solution is based on the overlay network topology abstracting each pair (IP domain + manager) with a node. The overlay network in this case is only a connectivity one, with no information about the resources available intra and inter-domain. Several alternative inter-domain paths are computes, at overlay level, for each destination domain. Then, the end to end QoS negotiation mechanism is used to reserve resources. Together they will act as a QoS inter-domain routing algorithm.

III. ENTHRONE END TO END QOS MANAGEMENT SYSTEM

As mentioned before the ENTHRONE project, IST 507637 (continued with ENTHRONE II, IST 038463) European project, cover the delivery of real time multimedia flows with end to end quality of services (QoS) guarantees, over IP based networks. To achieve this goal, a complex architecture has been proposed, which cover the entire audio-visual service distribution chain, including content generation, protection, distribution across QoS-enabled heterogeneous networks, and delivery of content at user terminals [2][3][4][5][6][7]. A complete business model has been considered, containing actors (entities) such as: *Service Providers (SP), Content Providers (CP), Network Providers (NP), Customers (Content Consumers – CC), etc.*

A. *Enthrone features*

ENTHRONE has defined an E2E QoS multi-domain *Enthrone Integrated Management Supervisor (EIMS).* It considers all actors mentioned above and their contractual service related relationships *Service Level Agreements (SLA)* and *Service Level Specifications (SLS)* as defined in [2][3][4][5][6][7]. One of the main EIMS components is the service management (SM). It is independent of particular management systems used by different NPs in their domains, and it is implemented in a distributed way, each network domain containing Service Management entities. It is present in SP, CP, NP CC entities, depending on the entity role in the E2E chain. The SM located in NPs should cooperate with each IP domain manager and also with other actors in the E2E chain.

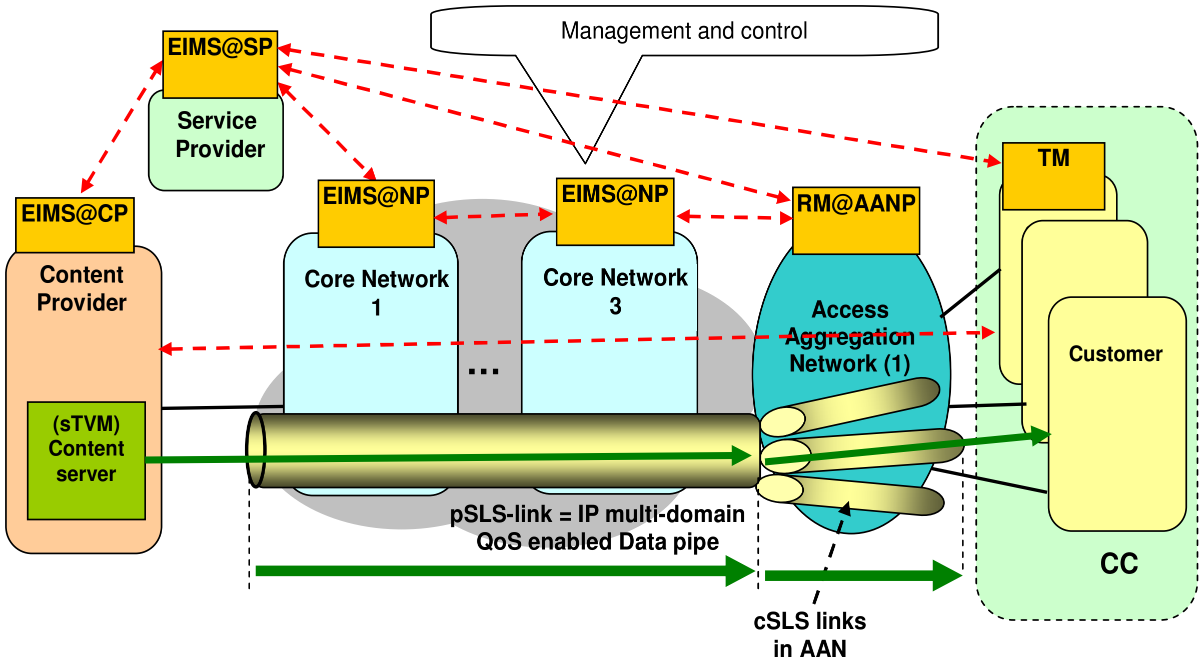

Figure 1. Forwarded cascaded model for pSLS negotiation

Legend:

EIMS@CP, EIMS@SP, EIMS@NP – ENTHRONE Integrated Management Supervisor at – respectively: Content Provider (CP) , Service Provider (SP) and Network Provider (NP)

sTVM –Source TV and Multimedia Processor (Content server)

AAN – Access Aggregation Network; AANP – AAN Provider

RM@AANP – Resource Manager of AAN (it is ENTHRONE compliant)

TM Terminal Manager

ENTHRONE supposes a multi-domain network composed of several IP domains and access networks (AN) at the edges. The CPs, SP, CCs, etc. are linked to these networks. The QoS transport concepts of ENTHRONE are shortly described below.

First, QoS enabled aggregated pipes, based on forecasted data, are established in the core network, part of the multi-domain network. They are logical pipes built by the Service Management entities. The aggregated QoS enabled pipe, called pSLS pipe, is identified by the associated pSLS agreement (Provider SLS) established between the Network Providers, in order to reserve the requested resources. Each pSLS-link belongs to a given QoS class, [20].

Then, slices/tracks of pSLS-links are used for individual flows based on individual cSLA/SLS contracts. An individual QoS enabled pipe is identified by a cSLS agreement, which is established between the manager of a Service Provider (EIMS@SP) and a CC for reserving the necessary resources for the requested quality of service. Several cSLSs pipes are aggregated at the core network level into an aggregated pSLS pipe.

In the data plane of core IP domains, Diffserv or MPLS can be used to enforce service differentiation corresponding to the QoS class defined. In the ANs, the traffic streams addressed to the users (Content Consumers) is treated similar to the *intserv*, i.e. individual resource reservations and invocations are made for each user.

B. Service Management at Network Provider

The EIMS architecture at NP (EIMS@NP) contains four functional planes: the *Service Plane (SPl)* establishes appropriate SLAs/SLSs among the operators/ providers/customers. The *Management Plane (MPl)* performs long term actions related to resource and traffic management. The *Control Plane (CPl)* performs the short term actions for resource and traffic engineering and control, including routing. In a multi-domain environment the *MPl* and *CPl* are logically divided in two sub-planes: inter-domain and intra-domain. Therefore, each domain may have its own management and control policies and mechanisms. The *Data Plane (DPl)* is responsible to transfer the multimedia data and to set the DiffServ traffic control mechanisms to assure the desired level of QoS.

The main task of the EIMS@NP is to find, negotiate and establish a QoS enabled pipe, from a Content Server (CS), belonging to a Content Provider, to a region where potential clients are located. Each pipe is established and identified by a chain of pSLS agreements, between successive NP managers. The forwarded cascaded model is used to build the pSLS pipes [5]. The pipes are unidirectional ones. An E2E negotiation protocol is used, [5] to negotiate the pSLS pipe construction across multiple network domains.

The process of establishing a pSLS–link/pipe is triggered by the SP. It decides, based on market analyses and users recorded requirements, to build a set of QoS enabled pipes, with QoS parameters described by a pSLS agreement. It starts a new negotiation session for each pSLS pipe establishment. It sends a pSLS Subscribe request to the EIMS@NP manager of the Content Consumer network domain. The EIMS@NP manager performs the QoS specific tasks such as admission control (AC), routing and service provisioning. To this aim, it splits the pSLS request into intra-domain respectively inter-domain pSLS request. It performs intra-domain routing to find the intra-domain route for the requested pSLS, and then it performs intra-domain AC. If these actions are successfully accomplished, and if the pSLS pipe is an inter-domain one, then the manager uses the routing agent to find the ingress point in the next domain, and send a pSLS Subscribe request towards the next domain. This negotiation is continued in chain, up to the destination domain, i.e., the domain of the CC access network. If the negotiation ends successfully, the QoS enabled pipe is considered logically established along the path from source to destination.

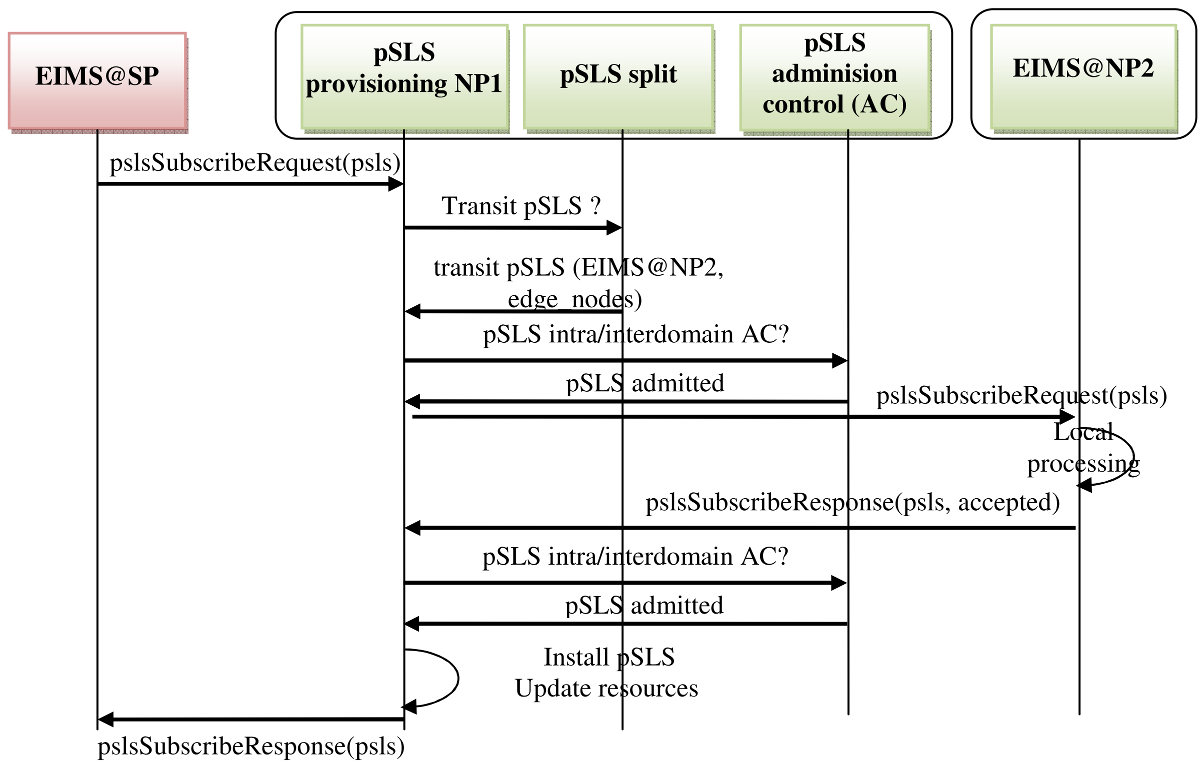

Figure 2. pSLS negotiation for QoS enable path establishment

The actual installation and configuration of routers is considered in ENTHRONE a separate action, and is done in invocation phase in a similar signaling way, plus the "vertical" commands given by EIMS@NP to the intra-domain resource manager.

After the pSLS pipe is active (i.e. subscribed and invoked) the Service Provider is ready to offer the new service to the users from the access network situated at the end of the pipe. Now the process of cSLS individual agreements establishment, for this new pSLS pipe, could be started.

IV. FINDING AN END TO END PATH WITH GUARANTIED QOS

A. General considerations

The main concepts of ENTRONE as stated in [8] are:

- E2E QoS over multiple domains is a main target of EIMS.
- But each AS has complete autonomy, regarding its network resources, including off-line traffic engineering (TE), network dimensioning and dynamic routing.
- Each Network Service Manager (cooperating with Intra-domain network resources manager) is supposed to know about its network resources in terms of QoS capabilities. ENTHRONE assumed that each AS manager has an abstract view of its network and output links towards neighbors, in a form of a set of virtual pipes (called Traffic Trunks in ENTHRONE I, see [5][6]), each such pipe belonging to a given QoS class.

A solution to this problem is to define/use routing protocols with QoS constraints, called QoS routing protocols. They can find a path between source and destination satisfying QoS constraints.

While finding the QoS path is only a first step, then maintaining the QoS with a given level of guarantees during the data transfer requires additional actions of resource management, including AC applied to new calls.

EIMS@NP management system performs these tasks. It is a centralized manager knowing the topology and resources of a domain. Being a central management node for a network domain, a centralized QoS routing solution is appropriate inside the domain.

On the other side, the multiple domain pSLS-links should also belong to some QoS classes and, therefore, inter-domain QoS aware routing information is necessary to increase the chances of successful pSLS establishment when negotiating the pSLSes. Several approaches are possible and they are summarized in [5]:

- NPs advertise their QoS capabilities with their associated scope through different methods (from automated peer-to-peer processes down to conventional techniques). A NP manager can locate and find out the QoS-classes offered by other domains (QoS capabilities, capacities, destination prefixes and costs).
- NPs implement a small number of well-known QoS classes. Inter-domain QoS services are created by constructing paths across those domains that support a particular QoS class. The BGP information is used to find destination prefixes. But QoS capabilities, capacities and costs, can be determined during pSLS negotiations – which may be successful or not.
- NPs advertise their QoS class capability and reachability through a protocol. Inter-domain QoS services are then created by constructing paths (which may not necessarily be the BGP path) across those domains that support a particular QoS class. This is path advertisement through a protocol.

B. The proposed overlay inter-domain QoS path finding solution

We proposed a simplified version [1], which takes into account the following assumption regarding the specific characteristics of the Enthrone system:

- The number of E2E QoS enabled pipes is not very large because they are long term aggregated pipes.
- The number of NP entities is much lower than the number of routers
- The EIMS@NPs are implemented on powerful and reliable machines, having enough computing and storage capabilities.
- The inter-domain core IP topology is rather stable and fixed; new elements are added at large time intervals.

This solution is also based on the idea of Overlay Virtual Network (OVN) [13], but in the first approach of our case, the OVN consists only of network domains (autonomous systems) abstracted as nodes. Each node will be represented by an EIMS@NP in this Overlay Virtual Network. This virtual network contains only information on connectivity between the domains, represented by the EIMS@NP nodes, or additionally static information regarding the inter-domain QoS parameters: links bandwidth, maximum jitter and delay, mean jitter and delay, etc.

This virtual connectivity topology (VCT) can be learned statically (offline) or dynamically.

The statically approach considers that the OVCT is built on a dedicated server – a topology server, like in the Domain Name Service (DNS). When a Network Provider wants to enter in the Enthrone system, then its EIMS@NP should register on this topology server. The topology server will return the Overlay Virtual Connectivity Topology. So, we will consider that each EIMS@NP has the knowledge of this connectivity topology. In the dynamic case each EIMS@NP, if wanting to build the OVCT, will query its directly linked (at data plane level) neighbor domains. It is supposed that it has the knowledge of such neighbors.

Each queried EIMS@NP returns only the list of its neighbors. At receipt of such information, the queerer EIMS@NP updates its topology data base (note that this process is not a flooding one as in OSPF). Then it queries the new nodes learned and so on. The process continues until the queerer node EIMS@NP learns the whole graph of "international" topology.

As we mentioned above the graph contains as nodes the EIMS@NP, which means that is made from the Network Service Managers of Enthrone capable domains.

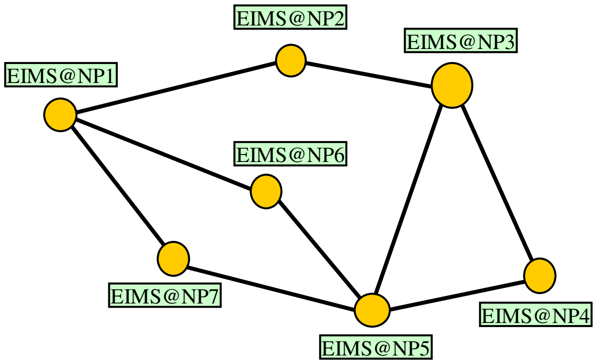

Figure 3. Overlay Virtual Connectivity Network

If the Enthrone system will be implemented at large scale, the number of nodes in the graph will be large, which means that the time required calculating the routing table will be also large. But because the topology changes events (adding new EIMS domains) are sparse ones (weeks, months), the topology construction process could run at large time intervals (once a day for example). In this case the routes calculation is triggered also at large time intervals, which means that it is enough time to determine the overlay paths. Another consequence is that the messages used to build the OVCT will not overload significantly the network. Enthrone capable domains can be separated by normal domains, with no Enthrone capabilities. In this case we consider that static QoS enabled pipes, are built between Enthrone capable domains, pipes crossing the Enthrone non capable domains. These domains (Enthrone non capable) will be transparent for the Enthrone domains.

On the graph learned, each EIMS@NP can compute several paths between source-destination pairs, thus being capable to offer alternative routes to the negotiation function.

The number of hops is used as a primary metric for the path choosing process. By the “hop” term we refer to a node in the Overlay Virtual Topology.

The process of route selection is as follows:

- When a request for a new pSLS arrived at one EIMS@NP, this will select the best path to the destination (the next EIMS@NP node that belong to this path), based on the overlay routing table.
- After the next hop is selected, the EIMS@NP will check if it has an intra-domain QoS enabled path for this route, i.e., between an appropriate ingress router and an egress router to the chosen next hop domain. If there is no such QoS enabled route, the next hope EIMS@NP node is selected from the overlay routing table.
- In case that, in the intra-domain, it is found a QoS enabled route, the EIMS@NP, based on mechanisms defined in Enthrone, trigger a request for a new pSLS negotiation to the chosen EIMS@NP neighbor.
- This process continues until the destination is reached. If the negotiation ends with success, than the pSLS pipe with guaranteed QoS parameters is found. If the process fails, then the EIMS@NP will choose another overlay path to the destination, and will start a new negotiation.

In Figure 4 the messages sequence for pSLS negotiation process, in the case of multiple paths towards the destination, is shown. The Service Provider decides to build a pSLS enable pipe between a source, located in the NP1 domain, and a destination, located in NP5 domain. We consider for this example that the working overlay topology is the one given in Figure 3. One can see that there are four possible routes between NP1 and NP5 domains. The first two of them, in terms of cost value, are the routs through NP6 and NP7 respectively. In Figure 4 it is illustrated the case when the pSLS negotiation along the route NP1-NP6-NP5 fails, due to admission control rejection by NP6 domain, either on intradomain pipe inside NP6 domain, or on the interdomain pipe between the NP6 and NP5 domains.

When it receives the rejection response at the pSLS subscription request the NP1 domain checks for an alternate route towards the NP5 domain. It finds the route through the NP7 domain, and starts a new negotiation using this new route. This negotiation ends successfully, so the QoS enable pipe between NP1 and NP5 will follow the route NP1-NP7-NP5.

This solution has the advantage of being simple, and that it not require at an AS the knowledge of current traffic trunks for the other network domains as in [13].

A drawback of our solution (proposed above) is a larger failure probability in negotiating a segment (therefore a longer mean time for negotiation process), if comparing with solutions which calculate the QoS path before the negotiation process. The latter approach increases the probability that the negotiation finished with success at the first try.

The path finding process described above is not based on BGP information at all. BGP is used only for best effort traffic. The process of QoS routing takes place at service management level. But it is possible in principle to use such BGP information.

V. DESIGN DETAILS

A. *Routing tables*

As mentioned before this solution is based on the knowledge of the overlay network connectivity topology. The topology can be kept in a form of a square matrix. The dimension M is equal to the number of nodes in the overlay topology network. Each entry r_{ij}, has an integer value. A zero value means that there is no direct connectivity between the nodes i and j. A value different from zero, value 1 for example, implies that there is a direct connection between the two nodes:

$$r_{ij} = \begin{cases} 1 & if\ \exists\, L_{ij} \\ 0 & if\ not\ \exists\, L_{ij} \end{cases} \quad (1)$$

L_{ij} represents the link between nodes i and j. Because the matrix is a sparse one, it can be easily compressed in order to be stored, in case that the dimension M is large.

Based on this overlay topology each EIMS@NP builds a routing table which contains, for each destination node in the network, the several possible paths to this destination node, and the costs associated with each of these paths. Because in the routing table several entries will exists for each destination, the QoS negotiation process will be able to be carried successively on multiple paths, increasing the probability that a path fulfilling the QoS requirements to be found.

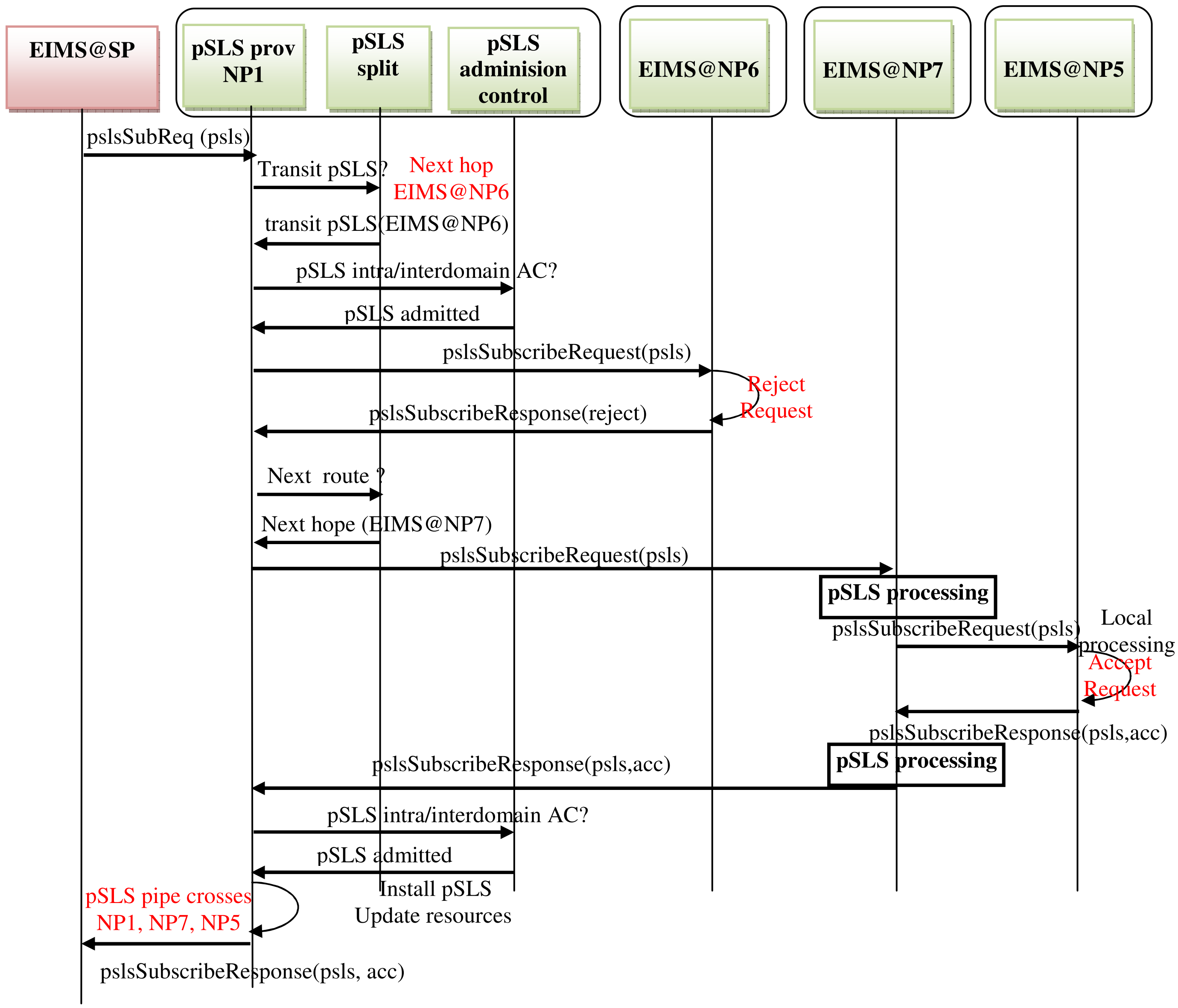

Figure 4. Overlay Virtual Connectivity Network

Because the number of possible paths from source to a certain destination could be high, we have limited it to the first four ones, with the lowest costs. If the neighbors number are less than four, than the number of possible routes towards a destination is limited to this number. It is used the same principle as in the case of distance vector protocols. In the case when there are several paths to the same destination EIMS@NP node, using as first next hop the same node, in the routing table it will be stored the best cost of all the possible paths going through that node.

This is not a limitation because in our case the routing decision is taken hop by hop so the source node has no idea what route to the destination will be chosen at the node where the paths are splitting. An EIMS@NP does not need to keep the whole path information (but the total cost only) because it cannot influence the route chosen decision at the next hops along the path.

Let's suppose that the EIMS@NP$_k$ node has the neighbor nodes EIMS@NP$_m$, EIMS@NP$_n$, EIMS@NP$_p$. The routing table from EIMS@NP$_k$ node to EIMS@NP$_l$ node will be:

TABLE I. ROUTING TABLE AT NODE *K* FOR NODE *L* DESTINATION

Destination	EIMS@NP$_l$	EIMS@NP$_l$	EIMS@NP$_l$
Nex Hop	EIMS@NP$_m$	EIMS@NP$_n$	EIMS@NP$_n$
Cost (Nb of hops)	5	0	3

The EIMS@NP at node *k* builds such a record for each node in the overlay network. This process, of searching several possible paths for each possible destination, in this overlay network topology, is an expensive one in terms of calculation. But based on the assumptions presented above, which are realistic ones, if such a management system will be implemented in the network domains, this routing table building process will be run only on topology updates, which means at very long time intervals. Such a process will put low computing overhead on the Service Manager. Also, it could be scheduled to run on intervals with low management activity [5]. Taking this in consideration, it could be considered that the routing table is a static one, and the route search process reduces to a simple database search one. It does not need to run the searching algorithm for each pSLS subscription request. It is enough to search, in the routing table, the route with the smallest cost, and forward the request to the chosen next node. If the negotiation for QoS parameters along this path failed, then it will chose the next path, in terms of cost, from the routing table.

B. Possible improvements

It is said that the solution did not take into consideration any QoS parameters, in the first phase, for path building process. This task left for the QoS negotiation process.

A possible improvement is to take into account some general data about the QoS parameters, in the path finding phase. For example, based on agreements with Service Managers of some domains, or based on some general QoS parameters of the domains, the Policy Based Management module could associate different costs for the links in the topology matrix. It is supposed that domains agreed to share these parameters, such as: the min/mean/max delay and jitter, introduced by the domain. In such a way the Policy module could influence the routing decision process. In this case the matrix element r_{ij} could be expressed as in (2):

$$r_{ij} = \begin{cases} c_{ij} & if\ \exists\, L_{ij} \\ 0 & if\ not\ \exists\, L_{ij} \end{cases} \qquad (2)$$

The value c_{ij} is the cost for the link L_{ij}, and could be established by weighting appropriately the general QoS parameters mentioned above. These weights could be established by the domain administrator and transmitted to the Policy module.

Also the cost of a link could be modified based on statistics regarding the acceptance or rejection rate of previous negotiated pSLS pipes. For example, if some domain with a good link cost rejects several times our requests we could modify the costs of the links crossing that domain.

Also, when the path cost is computed, it could be taken into account the existence of resource price agreements between some domains. These agreements could be negotiated using pull model, based on some statistics. For example, an EIMS@NP node has two different paths towards a destination with similar path costs. It chose the path with a better cost, but it also could periodically request resource price information from both neighbor nodes crossed by the two paths. If the second node has available resources and is interested to carry traffic from the source domain, it will propose a better resource price as a response to resource price requests. So the EIMS@NP source node could modify the routing table by improving the path cost for the second path, and the future pSLS pipe requests will be routed through the second path. Such a resource price communication could be easily implemented because the EIMS@NP managers are built as web-services, which implies very flexible communication capabilities.

C. Overlay topology building

For our solution we have chosen to build the overlay topology by means of successive interrogations of all the available nodes. The node, which decides to build/refresh the overlay topology, starts to interrogate all the other overlay nodes about their neighbors. It starts with its direct connected neighbors, and then continues interrogating the new found neighbors, and so on.

For the EIMS@NP implementation we have used the webservice technology. The interfaces between the EIMS@NP modules are implemented using WSDL language. The interdomain path finding WSDL interface it is used by EIMS@NP to interact with other EIMS@NPs, in order to build the overlay topology.

The inter-domain path finding WSDL interface has defined the following messages:

- ***getEimsNeighborsRequest*** ()
- ***getEimsNeighborsResponse***(EimsNeighborsArray eimsNeighbors)

- ***getDomainQoSRequest*** ()
- ***getDomainQoSResponse***(DomainQoS qos)

The first two messages are used by the Overlay Path Building module from EIMS@NP subsystem to build the overlay topology. The response message contains an array with all the neighbors of the interrogated domain, and their associated data about the webservices addresses, identification, and IP addresses.

The next two messages are used to get general information about the QoS parameters of the domain: min/max/mean delay and jitter, mean transit cost, max bandwidth. These values refer to the transit parameters for the domain. We have considered that such information could be offered by each domain without affecting its confidentiality policy. These parameters are used to establish the cost associated with a link between two neighbor domains. For establishing the cost we have weighted the normalized values for these parameters. The weights were chosen arbitrarily, such as their sum to be one. No studies have been done to find the optimal weights values.

The format of messages parameters are given in table 2.

TABLE II. DATA TYPE SECTION FOR THE INTERDOMAIN PATH FINDING WSDL INTERFACE

```
<wsdl:types>
     <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://webservice.enthrone.org/eims/
                   /InterdomainPath/datatype”>
<xsd:complexType name="EndPoint">
<xsd:sequence>
        <xsd:element name="IPAddress" type="xsd:string"/>
         <xsd:element name="NetMask" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Neighbor">
<xsd:sequence>
            <xsd:element name="id" type="xsd:string"/>
    <xsd:element name="OverlayPathWebserv" type="xsd:string"/>
             <xsd:element name="Address" type="tns:EndPoint "/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DomainQoS">
<xsd:sequence>
               <xsd:element name="minDelay" type="xsd:int"/>
               <xsd:element name="maxDelay" type="xsd:int"/>
              <xsd:element name="meanDelay" type="xsd:int"/>
              <xsd:element name="minJitter" type="xsd:int"/>
              <xsd:element name="maxJitter" type="xsd:int"/>
              <xsd:element name="meanDelay" type="xsd:int"/>
               <xsd:element name="meanCost" type="xsd:int"/>
             <xsd:element name="maxBandwidth" type="xsd:int"/>
</xsd:sequence>
</xsd:complexType>
</xsd:schema>
</wsdl:types>
```

In order to be able to perform the pSLS negotiation and to obtain the overlay topology, we have defined several database tables used to store the data required by the above mentioned operations. These tables are shortly described next:

- **Overlay_topology** table – it contains data about each EIMS node in the topology, such as the addresses of the web-services available, the IP address, the domain identifier, and the QoS parameters. It is updated by the Inter-domain Overlay Path module at each overlay topology building cycle. It is used by the overlay routing process to build the overlay topology matrix, used in the overlay route searching process.
- **Eims_neighbors** table – stores information about the neighbors for each EIMS node contained in the *overlay_topology* table. It is also updated by the Inter-domain Overlay Path module, at each overlay topology building cycle.
- **Overlay_interdomain_routes** table – is used to store several alternative routes towards a destination overlay node. The number of alternative routes is limited to four. It is managed by the overlay routing process.
- **Local_eims** table – stores information about the local NetSrvMngr@NP such as: IP address, web services ports, domain Id. It is managed by the system administrator.
- **Border_routers** table – stores informations about the local domains border routers. It contains the border routers IP address, and neighbor EIMS@NP reached through this border router. It is managed by the system administrator.
- **Access_networks** table – stores informations about the access networks for the local domain. It contains the access network IP address and the border router IP address. It is managed by the system administrator.
- **Local Eims_neighbors** table - stores information about the eims neighbors for the local domain. It contains information about the border routers used to connect the local domains and the neighbors, border router IP address, web service port addresses, etc. It is managed by the system administrator.
- **Domain_qos_parameters** table – it is used to store global QoS parameters about the domain. It is managed also by the system administrator.

D. *Functionality tests*

This solution was implemented on the test-bed build at our university in the Enthrone project framework [21] [22]. The test-bed consists of three Autonomous Systems, each managed by a Network Service Manager (EIMS@NP). The EIMS@NP managers are implemented using web services technology. Between domains the BGP protocol is used to route the best effort traffic. A Network Manager is used to install the pSLS pipes on network devices. Also the test-bed has a Service Provider EIMS Manager, and the other modules required by the Enthrone system. The connectivity tests involved only the Network Provider managers and Service Provider manager.

The EIMS@SP was used to trigger pSLS subscribe requests, between a Content Provider and one of the available Access Networks, until the resources on the lowest cost path between the chosen source and destination, were exhausted. Then, we triggered additional requests between the same source and destination. These new requests were admitted but the pSLS pipes were built along the next cheapest path between the chosen end points.

Because the test bed is a small one, is difficult to evaluate the performances of the proposed solution for a large number of domains. We have measured how fast a request for getting the neighbors EIMSs from a network domain is served. We have obtained a mean time less than 0.1s per request. If we take for example a topology consisting of 1000 domains then, because we can consider that the total processing time is increasing linearly with the number of domains, the total processing time requires to obtain the overlay topology is about 100s. We can increase it with 50% to take into account that at a large number of domains the local processing time, between two interrogations, could be higher. So, we could consider that for 1000 domains the topology building process takes about 150s, which is an acceptable value. Also, the solution used to build the overlay topology, implies a large number of messages to be exchanged in order to build the topology. Each node should communicate with the other nodes. But the messages exchanged are small, because each of them contains only a few data about the neighbors of the interrogated node. If it have been adopted a link state like protocol to build the topology, then the messages would have been much bigger, in case of large number of domains, so the amount of signaling data in the network would have been bigger too. Also, in our case we don't have convergence problems.

It has not been evaluated till now the time needed to compute several paths towards all the destinations nodes in the overlay topology.

The test bed used is not appropriate to test the scalability for the path finding process performed in the first phase. It was only used to see that the routing table is built correctly, containing several paths towards each destination domain in the topology. Then, several requests for QoS enabled pSLS pipes were triggered. These pipes were built along the first path specified in the routing table. When the resources on this path were exhausted, during the negotiation process, the next route was used for the following pSLS pipe. These tests proved that the solution is able to find QoS enabled pipes, in a multi domain environment.

VI. ESTABLISHMENTCONCLUSION AND FUTURE WORK

It has proposed a simple solution for solving the problem of QoS enabled inter-domain path finding, in the presence of a Network Service Management system, capable of QoS enabled pSLS pipes negotiation.

Because it does not require at a domain the knowledge of other domain resources, it could be accepted by the actual network providers. Another advantage is, that it does not burden a given domain manager with the need of knowing the available traffic trunks of other network domains. Also, by separating the process of path finding from the QoS negotiation, the path searching process does not need to work real time. So we can find several paths in very complex overlay topologies. By simplifying the overlay topology, considering only the domain managers as topology nodes, our solution will work for very complex topologies, being no need for a hierarchical approach.

The solution has the main disadvantage that it does work only in the presence of a QoS negotiation system capable. It is based on this feature to check the QoS constraints along the paths founded in the overlay topology. Another disadvantage is that, it may not find the best QoS enabled path, as could be the case with other solutions.

But it is simple, and is well suited for ENTHRONE Integrated Management System. The solution is also naturally extensible for more sophisticated techniques in QoS capable paths finding.

Further studies and simulations will be done in order to validate this solution for a real network environment. Also, it has been suppose that, because the path finding process could be run offline, and the topology is a simplified one, a non hierarchical solution could be adopted for Internet. Simulations should be done to establish the amount of resources need by such a process.

ACKNOWLEDGMENT

This work was supported by the FP6 project ENTHRONE.

REFERENCES

[1] S.G. Obreja,, E. Borcoci, "Overlay Topology Based Inter-domain QoS Paths Building". AICT apos;08. Fourth Advanced International Conference on Telecommunications, Volume , Issue , 8-13 June 2008 Page(s):64 – 70. Greece, Athens.

[2] ENTHRONE I Deliverable 05 "IMS Architecture Definition and Specification", June 2004.

[3] A. Kourtis, H. Asgari, A. Mehaoua, E. Borcoci, S. Eccles, E. Le Doeuff, P. Bretillon, J. Lauterjung, M. Stiemerling, "Overall Network Architecture", D21 ENTHRONE Deliverable, May 2004.

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

[5] 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

[6] P.Bretillon ed., et. al, ENTHRONE II Deliverable D01: Overall system architecture – version 2, 2007

[7] P.Souto ed., et al, ENTHRONE II Deliverable D03f: EIMS for ENTHRONE 2

[8] E. Borcoci, Ş. Obreja eds., et al, ENTHRONE II Deliverable D18f: Service Management and QoS provisioning.

[9] Project P1008, "Inter-operator interfaces for ensuring end-to-end IP QoS", Deliverable 2, Selected Scenarios and requirements for end-to-end IP QoS man-agement, January 2001.

[10] 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 Ser-vices in MPLS-Based Networks", IEEE Comm. Magazine, May 2001, pp. 80-88.

[11] E.Marilly et. al, "SLAs: A Main Challenge for Next Generation Networks", 2nd European Conference on Universal Multiservice Networks, ECUMN'2002 April 8-10, 2002.

[12] T.Engel, H.Granzer, B.F. Koch, M.Winter, P.Sampatakos I.S. Venieris, H.Hussmann, F.Ricciato, S.Salsano, "AQUILA: Adaptive Resource Control for QoS Using an IP-Based Layered Architecture", IEEE Communications Magazine, January 2003, pp. 46-53. See also http://www-st.inf.tu-dresden.de/aquila/

[13] Fabio L. Verdi, Maurıcio F. Magalhaes "Using Virtualization to Provide Interdomain QoS-enabled Routing", Journal of Networks, April 2007.

[14] Z. Li, P. Mohapatra, and C. Chuah, "Virtual Multi-Homing: On the Feasibility of Combining Overlay Routing with BGP Routing," University of California at Davis Technical Report: CSE-2005-2, 2005.

[15] Z. Wang and J. Crowcroft, "Quality of Service Routing for supporting multimedia applications", IEEE Journal of Selected Areas in Communication (JSAC), 14 (7) (1996), pp. 1228-1234.

[16] D. Eppstein, "Finding k-shortest paths", SIAM Journal on Computing, 28 (2) (1998), pp. 652-673.

[17] D. Griffin, J. Spencer, J. Griem, M. Boucadair, P. Morand, M. Howarth, N. Wang, G. Pavlou, A. Asgari, P. Georgatso, "Interdomain routing through QoS-class planes", Communications Magazine, IEEE, Feb.2007, Volume: 45, Issue: 2, page(s): 88-95.

[18] S.P. Romano, ed., "Resource Management in SLA Networks", D2.3 CADENUS Deliverable, May 2003.

[19] T.Ahmed, A.Asgari, A.Mehaoua, E.Borcoci, L.Berti-iquille, G.Kormentzas "End-to-End QoS Provisioning Through an Integrated Management System for Multimedia Content Delivery", Computer Communication Journal, Volume 30, Issue 3 (February 2007), pages 638-651.

[20] 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, Lisbon, Volume, Issue, 17-20 July 2005 Page(s): 46 - 52. Portugal, Lisbon.

[21] T. Ahmed, ed. et al., ENTHRONE II Deliverable D27 "Pilot and services integration and tests", March 2008

[22] ENTHRONE II Deliverable D28 "Trials and evaluation", November 2008