Optimization of Overlay QoS Constrained Routing and Mapping Algorithm for Virtual Content Aware networks

Radu Dinel Miruta, Eugen Borcoci
Telecommunication Dept.
University POLITEHNICA of Bucharest
Bucharest, Romania
radu.miruta@elcom.pub.ro, eugen.borcoci@elcom.pub.ro

Abstract—Multimedia Services including video distribution are increasingly required by the current market and will be also a target of the Future Internet. One method to customize the multi-domain guaranteed transport with several QoS classes of services is to create Virtual Content Aware Networks (VCAN) constructed as overlays over IP networks. The mapping of VCANs onto real multi-domain topologies is needed. This paper develops new optimizations to increase the performances of a previously proposed combined hierarchical multi-domain algorithm performing VCAN mapping with QoS constraints.

Keywords-Virtual Content-Aware Networking; Network Aware Applications; Multi-domain; QoS; Management; Constrained routing; Future Internet.

I. INTRODUCTION

The transport of media streams over heterogeneous IP networks is a need of the current and also Future Internet. However, assuring the quality of services (QoS) and other special needs of the high level services, (security, reliability, etc.) are still not answered satisfactorily by the current public networks, except the "wall gardened" networks, fully owned and controlled by operators (e.g., IPTV distribution networks).

One new solution, content-oriented, is to transport media flows over some previously created (on demand) Virtual Content Aware Networks (VCANs). They are usually constructed as overlays on top of IP level [1][4], based on (light) virtualisation techniques. In a multi-domain network and several operators context, the VCANs can be multiple-domain spanning, therefore several Network Providers (e.g., ISP) might cooperate towards this goal. Given that a VCAN is an overlay and the fact that NP/ISPs are independent entities, it is useful to define new business role, i.e., a new provider level called CAN Provider (CANP) [4][5][13]. This is capable to aggregate network resources offered by several NP/ISPs and to create VCANs on top of them. The VCANs are offered by CANPs to by High Level Services Providers (SP) which deploy media services for communities of users, or they are asked by the SPs to CANPs. Each VCAN can be associated to a given QoS class.

A VCAN solution to media flow dedicated transport is proposed in ALICANTE European FP7 ICT research project, "Media Ecosystem Deployment Through Ubiquitous Content-Aware Network Environments" [4]. The VCANs are realized as parallel data planes [10] and are content-type recognition capable under control of a single management and control – M&C plane. The solution, while not fully content oriented as in [2][3], is attractive because it can offer a possibility of seamless deployment and put much less processing tasks on the content aware routers than Information/Content Centric Networking (CCN/CCN) approach.

The network contains several Core Network Domains (CND) and access networks (ANs). The ANs are out of scope of ALICANTE and to VCANs; access network resource control is considered as a separate problem. The CNDs belong to NPs and can be Autonomous Systems (AS). The CAN layer Management and Control (M&C) is partially distributed: one *CAN Manager* (CANMgr) belonging to CANP exists for each IP domain, performing VCAN planning, provisioning, advertisement, offering, negotiation installation and exploitation. Each domain has an *Intra-domain Network Resource Manager* (Intra-NRM), which configures the network nodes. The EU terminals are connected to the network through Home Boxes (HB). The novel CAN routers are called *Media-Aware Network Elements (MANE)* to emphasize their additional capabilities: content and context – awareness. The CAN layer cooperates with HB and SE by offering them CAN services. In the CNDs DiffServ and/or MPLS technologies can support splitting the sets of flows in QoS classes (QC), with a mapping between the VCANs and the QCs with several levels of QoS granularities, [4][5]. The QoS behavior of each VCAN (seen as one of the parallel Internet planes) is established by the SP-CANP.

The VCANs asked from a CANP by an SP should be mapped onto real multi-domain network topology, while respecting some QoS constraints. This provisioning is done through negotiations performed between CAN Managers associated to each network domain. One CANMgr is the initiator of VCAN construction, at request of an SP. If necessary the initiator communicates with other CANMgrs, to finally agree a reservation and then a real allocation (i.e., installation in the network routers) of network resources necessary for a VCAN. A CAN Planning module inside each CANMGr is the entity which runs a *combined algorithm doing QoS constrained routing, VCAN mapping and resource logical reservation*. In this set of actions it is supposed that the initiator CANMgr knows the inter-domain topology at an overlay level and also a summary of each network domain topology, in terms of abstract trunks (e.g., {ingress, egress, bandwidth, QoS class, ..}). This knowledge is delivered by an additional discovery service and is out of scope of this article. Previous papers, of the same authors [5][13], developed and implemented the combined VCAN mapping algorithm. *This article continues the previous work by proposing several techniques for performance and scalability improvement.*

The paper organization is described below. Section 2 makes a short overview on samples of related work. Section 3 summarizes the original VCANs planning and mapping algorithm. Sections 4 and 5 contain the main contribution of this paper. Section 4 develops the optimization techniques, and Section 5 presents some performance analysis results. Section 6 contains conclusions and future work orientation.

II. RELATED WORK

The basic algorithm proposed in [5] and [13] by these authors has as goal to map customized QoS capable VCANs over several network domains, independently managed, to efficiently transport of real-time and media traffic. This paper proposes some optimizations of that basic algorithm. Therefore some related works presented previously are only summarized here. Given that generally such mapping problems are NP-hard [12], a convenient solution has been selected to fit the ALICANTE architecture specific needs. In particular, the CAN Managers and Intra-domain Network Resources Managers– have knowledge on the status of their resources. After paths finding, a negotiation protocol is run, [4][5], between domain managers, to establish inter-domains SLAs. If no QoS constraints are used during routing there are significant chances that the SLA negotiation will fail. A better solutions is to first search for QoS enabled paths, as in [5][6][7][13], followed by SLS conclusions.

The Service Overlay Networks are discussed in [14] which are partially similar to our VCANs. The following assumptions have been considered - part of them similar to our VCAN case: pre-determined location of the overlay nodes; the overlay link metric is the delay; the overlay path between a pair of overlay nodes is selected by using the Dijkstra algorithm; each overlay path is composed of IP-layer links. At IP layer, the cost of each link is 1/Bandwidth, and the shortest path between a pair of IP nodes is computed by using the Dijkstra algorithm. Several overlay topologies have been studied: Full-Mesh (FMsh), K-Minimum Spanning Tree (KMST), Mesh-Tree (MT), Adjacent Connections (ACON), K-Shortest Path Tree (KSPT), Pruned Adjacent Connection (PAC) and Demand –aware adjacent connection (DAC). The overall optimization cost function is a weighted sum of delays on different overlay links weighted with the traffic demands between pair of overlay nodes. The “best” overlay topology was considered if, on equal terms of accepted traffic and performance, has the lowest overhead (minimum number of interfaces per/node), due to the overlay network maintenance traffic. This is not a primary criterion of our solution.

The algorithm in [15] considers both link capacity, and overlay servers capacities. However, this last parameter is out of the scope of our proposal.

The ALICANTE solution is similar to the K-Shortest Path Tree (KSPT), [14] in terms of topology. However, ALICANTE includes in the algorithm for VCAN mapping not only QoS constrained routing based on modified Dijkstra algorithm, but also resource reservation – thus supporting the QoS assurance.

Our solution assumes that an inter-domain overlay QoS peering and routing [13][14], has been solved in the sense that a topology discovery protocol and service exists, capable to make the CAN Mangers aware of topology (at overlay level) and capacities aware.

Overlay networks having QoS capabilities are described in several papers, [6][7][8][9][14]. The solution proposed in [5][13] has a new characteristic that it tries to combine in the same algorithm QoS enabled (constrained) routing, admission control, mapping and resource reservation for VCANs.

III. BASIC VCAN MAPPING ALGORITHM

This section summarizes the initial algorithm, [5][13], run by the CAN Manager/Intra-NRM in order to map VCAN QoS requirements onto physical network resources onto one or more core network domains (CND). The main input data will be: the multi-domain network graph (topology, capacities) - collected by the topology discovery service; Traffic Demand Matrix (TDM) - asked by SP to the initiator CAN Manager. Note that actually the SP request can contain more parameters in an SLA template including several aspects of the VCAN life and operation. We only considered here the relevant parameters for the VCAN mapping algorithm. The output of the algorithm will be the mapping of TDM on real paths after admission control is done to check respecting the minimum bandwidth constraints and also optimize the network resource usage.

The CANMgr/Intra-NRM runs a combined constrained routing, mapping and admission control and resource reservation algorithm. The metric proposed for a link, is selected as to lead to selection of the widest path.

The cost of a intra-domain link (i,j) in the overlay graph is defined as additive metric $C(i,j) = Breq/Bij = Breq/Bavail$, where Bij is the available bandwidth on this link and *Breq* is the bandwidth requested for that link, [5][13]. The ratio also is seen as link utilization factor; that is the alternative notations will be used: $C(i,j) = U_{link\_ij}$. The constraint is: $(Breq/Bij < 1)$. Therefore in each action of path search the branches not satisfying this constraint should be not considered. The metric is additive, so one can apply modified Dijkstra algorithm to compute the *Shortest Path Trees (SPT),* one tree for each ingress node where the traffic flows will enter. Note that *Breq/Bij* can be only computed if we know the mapping TT - link (i.e., we know *Breq* for a given link), which is not yet our case. The mapping is to be done jointly with the routing process. So in the first approximation we consider 1/Bij as an additive link metric. Other more sophisticated metrics could be considered, e.g., including the delay, provided that this can be estimated/measured by a monitoring system.

The solution presented here is *valid for both unicast and multicast VCANs*; a multicast TDM is actually a particular case of a unicast TDM matrix. In unicast case the TDM entries are tuples including information like *(ingress, egress, bandwidth, ..)* where each egress may have a different bandwidth request. In multicast case the whole TDM is representing a tree or a set of trees where the bandwidth of a tree is the same for all egress points associated with a root (i.e., ingress of the TDM).

IV. VCAN MAPPING ALGORITHM OPTIMIZATION

One problem discussed in this section is how to reduce the complexity of calculus given that we may have large graphs in a multi domain topology. The Dijkstra's original algorithm runs in $O(|V|^2)$ complexity, or in the best case if the implementation is based on a min-priority queue implemented by a Fibonacci heap, then one has $O(|E| + |V| \log |V|)$ (Fredman & Tarjan 1984). In our case, a TDM may have *n* ingress points (lines), so the complexity is n* O(Dijkstra). For each computation (out of a total of n) the algorithm will determine a constrained Shortest Path Tree (SPT), and then will map the TDM hoses (each TDM line corresponds with a hose) on this SPT. The reservation is done by subtracting the requested capacities from the initial ones per each branch of the graph.

However, the order in which the hoses (i.e., requests) are analysed (and subsequent subtraction) may change the final result. Then if the CAN Manager wants the best VCAN mapping and least overall utilization, then it should check all combinations of computation. The most trivial solution is to recompute the step 2 of the algorithm for other order of inputs given by the bijective function $f(GR_1, ..GR_n) \rightarrow \{GR_{k1}, GR_{k2}, ..GR_{kn}\}$ which creates actually permutations of the set $\{GR_1, ..GR_n\}$, where each GR_k represents a group of requests (i.e., a hose) associated to an ingress point of traffic. The final mapping solution will be the one having the least overall utilization. The overall complexity will be *n*n!* O(Dijkstra)* which has not so good scalability, [13]. Acceptance of such a solution could exist however, given that VCANs are constructed for medium-long term and the frequency of SP requests for VCANs are rather low (non hard-real time computation).

A. Service Provider Driven Priorities

The order of analysis can be more deterministic and the number of computation reduced if the SP assigns a priority order to its requests; then less or even no permutations are needed. Note that a group of requests is represented by a tuple (ingress, egress1, egress2, …). The network available capacities are priority reserved in order, first for the most important requests. In ALICANTE context the SP is the appropriate business actor to know which traffic pipes of the TDM are more important.

In other contexts, the NP could create some particular rules for establish an honoring list. One possible rule could be that the request with the higher requested capacity to be solved the first one. However, in ALICANTE and not only, not always the higher capacity value signifies the most important request.

Two cases are for analysis: a. *strict monotonic row of groups_of_requests priorities;* b. *monotonic row of group_of_requests priorities* (i.e., some of them may be equal). In case *a.* the complexity will be reduced drastically, i.e., we have *complexity = n* O(Dijkstra),* given that the order is strictly determined. In case *b.* one has a structure: $\{(GR_{1,1}, GR_{1,2}, \dots GR_{1,n1}), (GR_{2,1}, GR_{2,2}, \dots GR_{2,n2}), \dots. (GR_{k1}, GR_{k2}, \dots GR_{k,nk})\}$, where, inside each set of a group (…), all requests have the same priority. Additionally we suppose that the priorities for groups are in strict decreasing order. We also have $n1 + n2 + \dots + nk = n$, i.e., the total number of requests. Still in this case one gets a serious reduction in number of computations, given that n1! + n2! + … is much less than *n!.*

As a simple example we suppose that n1= n2 = ..nk =n/k. In this case the total number of permutations will be k [(n/k)!]. Using Stirling approximation formula $n! \sim (2\pi n)^{1/2}(n/e)^n$, we get a reduction factor equal to

$$n!/[k^* (n/k)!] \sim (k)^{n-1/2} \tag{1}$$

For instance if we have n=10, k=2 we have a reduction factor in number of computation of ~714 and this increases rapidly with *n.* Therefore the solution is much more scalable for large network graphs.

B. Priority Specification Model

The proposed model in this section can be used for both VCAN mapping solutions (in one or two steps) presented in [5][13]. All requests from the received set are grouped based on the source node and *group priority* is defined (lower value means higher priority). In the case of several groups with the same priority, as shown in the sub-section above, the algorithm will permute the processing order obtaining the best cost. Note that the algorithm details have been already described in [5].

In the proposed algorithm, once a group is chosen for analysis, all the requests from that group are processed- but in which order? To offer a maximum flexibility solution w.r.t. SP interests, one should admit that SP can specify a priority for each individual request. So, the model will allow two levels of priorities: *per group* and *per request* inside the group. The choice here is that *group priority has precedence on the individual request one.* However in practice this is not always true. In such cases the solution is to define distinct groups for some requests, for which we want given priorities, despite that the ingress point is the same.

Fig. 1 shows an example on how a TDM can have a split of request in groups assigned to a given ingress point, where the individual requests may have different priorities inside each group. The values P represent the priorities of an individual request.

The future VCAN satisfying this TDM is represented as an outer circle. The actual network may have several interconnected Core Network Domains (three in our example).

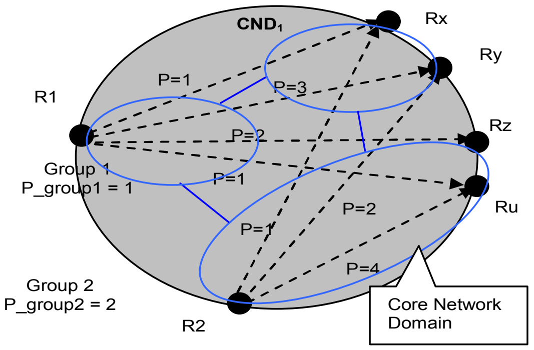
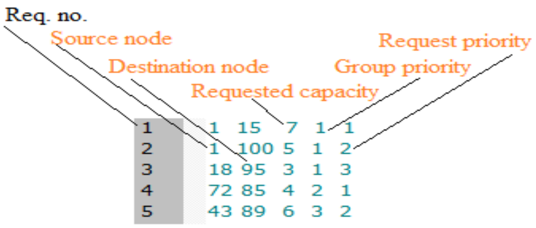


Figure 1. Example of prioritized requests for resources

The sequence of solving the requests in the above example is: 1. {R1-Rx, R1-Ru, R1-Rz, R1-Ry }; 2. {R2-Rx, R2-Ry, R2-Ru}.



1	1	15	7	1	1
2	1	100	5	1	2
3	18	95	3	1	3
4	72	85	4	2	1
5	43	89	6	3	2

Figure 2. Traffic Demand Matrix Example

Fig. 2 shows a simplified example of a TDM, containing several requests. This TDM is produced by SP and delivered to the CAN Manger initiating the VCAN construction. Each line of the matrix specifies an individual request as *{source node, destination node, requested capacity, group priority, individual request priority}*. The groups are associated with source nodes being {1, 18, 72, 43}. One can see that the groups {1, 18} have equal same priority =1. This will determine two permutations when analyzing the requests. Each individual request priority inside a group has only local significance.

V. PERFORMANCE ANALYSIS

This section will present simulation results. The basic algorithm implementation proposed in [5] has been upgraded to leverage priorities. The network has been simulated, by generating the topology using specialized tools.

A. *Simulation settings*

The tools have been a Network Analysis and Routing eValuation – NARVAL module 2.0.1-1 [16] from Scilab 5.4.0 [17] to generate complex multiple-domains network topologies. Fig. 3 illustrates a two level hierarchy where the bottom part represents the inter-domain network graph and the top one signifies the intra-domain one.

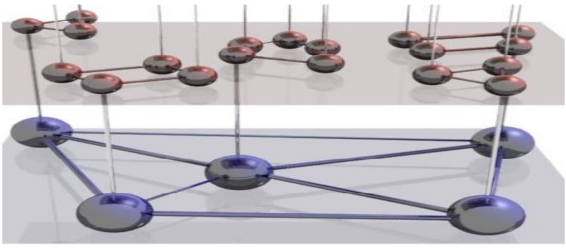


Figure 3. Two level hierarchy topology [16]

Creation of some scripts using the NARVAL module allowed constructing a large hierarchical network with backbone of inter-domains links; each node at inter-domain level represents an abstraction of an intra-domain topology. Each segment for both inter and intra-domain areas has an associated bandwidth generated in respect to a Gaussian distribution centered in 70.

Some default functions have been modified in order to obtain a two levels hierarchical topology (by default there are five levels). The network backbone of size *n* is assumed to be created based on the Waxman model [18], with parameters *a* and *b*. The largest connex subnetwork was extracted. As a matter of use case for the algorithm the backbone needs to be fully connected (through fully connected we understand that there are not isolated nodes, not that the topology is connected in a full mesh fashion). Thereafter the second layer was added according to the Waxman algorithm, too (the same parameters *a* and *b* are used for each network layer). New nodes are added by small groups of size randomly selected into the range [1, 2, 3, …, *nl*]; *cv* is a *s*-length vector, where *s* is the no of layers, that contains the colors used to display each layer [17]. The nodes of the first layer have a diameter (diameter of the circle from the figure representing a node) equal to *db*. The nodes diameter is constant for layer, but we reduce this value when we move to the next layer with a rate of *dd*. The network generated has with 27 backbone nodes and 116 intra-domain nodes; (total is 143 nodes). The topology obtained is presented in Figure 4.

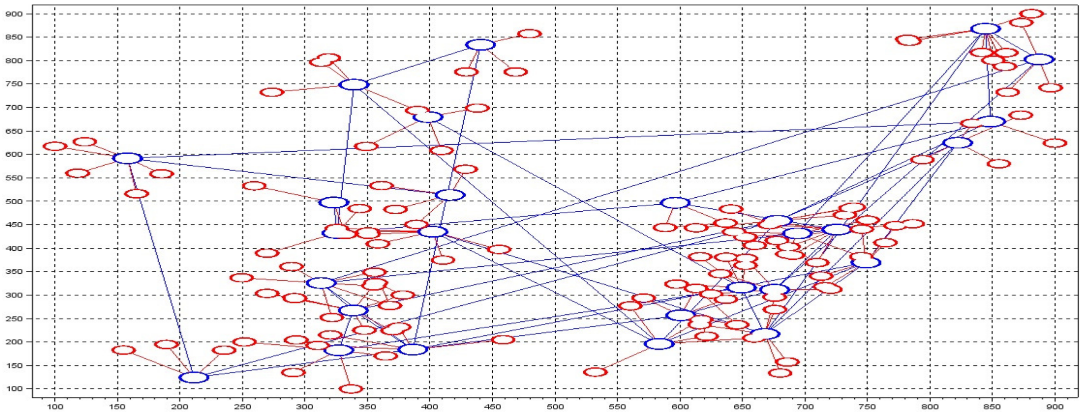


Figure 4. Two level hierarchical network topology generated with Narval tool

The topology was generated using the following parameters values:

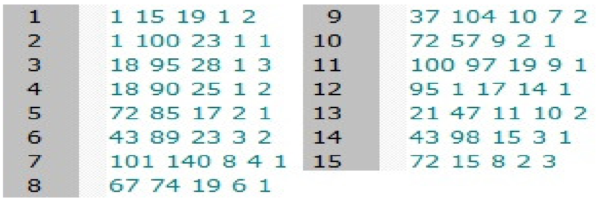
a=0.3;//first parameter of the Waxman model;
b=0.4;//second parameter of the Waxman model;
n=27;//network backbone size; l=1000;//network squared area side;
nl=7;//maximal quantity of nodes per subnetwork;
db=20;//original diameter of nodes;
dd=5;//diameter difference between successive network layers;
cv=[2 5];//color of each network layer.

Using some scripts the adjacency matrix has been extracted with values of 0 and 1 (0 means no link between nodes and 1 means the presence of a link). In order to obtain an adjacency matrix where the presence of a link is represented with the available bandwidth value instead of simply 1, as a last part of our simulations settings we assigned an element of a previous created weight vector (using a Gaussian distribution centered in 70) to all of the adjacency matrix elements different from 0. The weight vector contains elements to be assigned as bandwidth value to each existing link. This assignation process of the corresponding bandwidth value for each link can be seen below:

```
[l c]=size(AM);
ind=find(AM==1);//presence of a link
AMW=AM;//matrix with weight initialized with AM
for i=1:length(ind)
   il=modulo(ind(i),l);//line index
   if (il==0) then
      il=l;
   end
   ic=ceil(ind(i)/c);//column index
   AMW(il,ic)=g.edge_length(NARVAL_G_Nodes2Edge(g,il,ic));
end
```

B. Simulation Results

The TDM proposed contains a set of 15 requests divided into 9 groups with different priorities and different individual priorities as in Fig. 5.



1	1 15 19 1 2	9	37 104 10 7 2
2	1 100 23 1 1	10	72 57 9 2 1
3	18 95 28 1 3	11	100 97 19 9 1
4	18 90 25 1 2	12	95 1 17 14 1
5	72 85 17 2 1	13	21 47 11 10 2
6	43 89 23 3 2	14	43 98 15 3 1
7	101 140 8 4 1	15	72 15 8 2 3
8	67 74 19 6 1		

Figure 5. Example of a set of requests with priorities

Running the algorithm, it produces below results (for only *two permutations* in case of groups with the same priorities):

```
==============================
Input file Scilab1.in:
==============================
Request 1->100, load 23: 1 6 4 16 100
Request 1->15, load 19: 1 14 15
Request 18->90, load 25: 18 4 16 13 90
Request 18->95, load 28 unsatisfied on 18->4,avail.cap. 11. path traveled: 18 4 14 95
Request blind 18->95, load 28, cost 2.860116: 18 8 19 23 14 95
….
Request 37->104, load 10 unsatisfied. Node unreachable.

Cost: 28.76079 of which blind: 9.19815 Satisfied req: 13 / 15
-------------------------------------------
Request 18->90, load 25: 18 4 16 13 90
Request 18->95, load 28 unsatisfied on 18->4,avail.cap. 11. path traveled: 18 4 14 95
Request blind 18->95, load 28, cost 3.795075: 18 8 19 6 1 14 95
Request 1->100, load 23: 1 6 4 16 100
Request 1->15, load 19: 1 6 19 23 14 15
…
Request 43->89, load 23 unsatisfied on 4->14,avail.cap. 14. path traveled: 4 14 15 24 25 13 89
Request blind 43->89, load 23, cost 6.202275: 43 4 6 19 23 14 15 24 25 13 89
…
Cost: 31.10171 of which blind: 9.99735 Satisfied req:  13 / 15
-------------------------------------------
Best cost: 28.760790
Satisfied Requests: 13 / 15
Total time: 0.018000
```

As it can be seen, only 13 requests from 15 are solved and a better cost is associated to the first order (excepting the situation of node unreachable). Two requests could not be solved using the modified Djikstra algorithm and in this special case the blind search found an alternative path. Only this blind search adds an extra cost because of the longer found path compared to the Djikstra one. All requests are honored according to the group and individual priorities. As an alternative choice, in the case of many groups with the same priority, one can be specified how many permutations are desired. We used only 2 in this example.

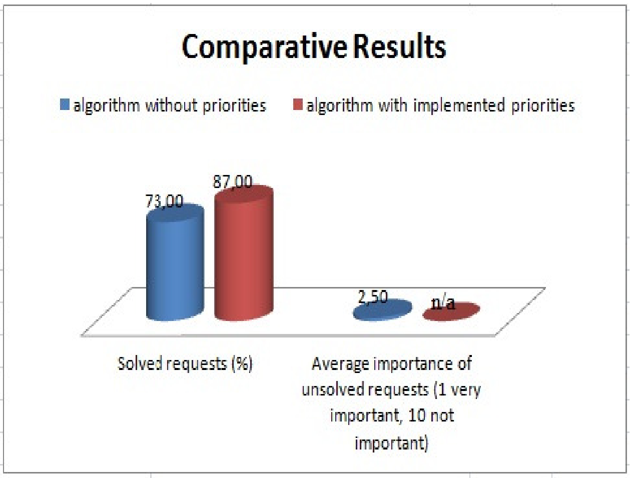


Figure 6. Comparative results

Running the basic algorithm without any priorities, but for the same input file, *only 11/15 requests are solved.* The most important thing is that some important requests (with priorities 2 and 3 in this new context) have not been solved, while and some less important requests have been solved. Fig. 6 shows comparative results for initial algorithm and that one having priorities.

For both cases, there were not taken into consideration the cases of unreachable nodes. Even if the network graph is constructed as a connex one, because of some optimization techniques used during the implementation (removing from the existing graph all segments which do not respect the condition: *available bandwidth >= minimum request* bandwidth value from the group) some nodes could become unreachable. For the case with prioritized requests the two unsolved requests are because of the unreachable node, so in this comparison we consider unsolved requests as n/a.

VI. CONCLUSIONS

This paper proposed optimization methods to increase the performances of a previously developed combined algorithm, having the goal to map Virtual Content Aware Networks on top of multi-domain IP topologies, while respecting QoS constraints. It is shown how introduction of priorities in the Traffic Demand Matrices asked by the Service Provider can greatly reduce the number of computations while increasing the number of solved requests, in comparison with the basic algorithm. Future work will extend the evaluation on several types of topologies (sparse, dense) and allocate resources for several types of QoS classes. Currently, the algorithm is developed inside the CAN Manger of the ALICANTE FP7 project.

ACKNOWLEDGMENTS

This work was supported partially by the EC in the context of the ALICANTE project (FP7-ICT-248652) and partially by the project POSDRU/88/1.5/S/61178.

REFERENCES

[1] T. Anderson, L. Peterson, S. Shenker, and J. Turner, “Overcoming the Internet Impasse through Virtualization”, Computer, vol. 38, no. 4, Apr. 2005, pp. 34–41.

[2] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, A Survey on Content-Oriented Networking for Efficient Content Delivery, IEEE Comm. Magazine, March 2011.

[3] V. Jacobson et al., “Networking Named Content,” CoNEXT ’09, New York, NY, 2009, pp. 1–12.

[4] FP7 ICT project, “MediA Ecosystem Deployment Through Ubiquitous Content-Aware Network Environments”, No248652, “D2.1: ALICANTE Overall System and Components Definition and Specifications”, http://www.ict-alicante.eu/ (last access March 2013)

[5] R. Miruta, E. Borcoci and E. Palis „Planning and Provisioning of Virtual Content Aware Networks over IP Infrastructures”, International Conference on Telecommunication and Multimedia, July 2012, pp. 118-123

[6] Z. Li, P. Mohapatra, “QRON: QoS-Aware Routing in Overlay Networks”, IEEE Journal on Selected Areas in Communications, Vol. 22, No. 1, January 2004, pp.29-39.

[7] J. Galán-Jiménez and A. Gazo-Cervero, “Overview and Challenges of Overlay Networks”, International Journal of Computer Science & Engineering Survey (IJCSES) Vol.2, No.1, Feb 2011, DOI : 10.5121/ijcses.2011.2102 19

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

[9] L.F. Verdi and F. Magalhaes “Using Virtualization to Provide Interdomain QoS-enabled Routing”, Journal of Networks, April 2007, pp. 23-32.

[10] M. Boucadair, et al., “A Framework for End-to-End Service Differentiation: Network Planes and Parallel Internets”, IEEE Communications. Magazine, Sept. 2007, pp. 134-143.

[11] Z. Wang and J. Crowcroft, “Quality-of-service routing for supporting multimedia applications”, IEEE Journal on Selected Areas in Communications, vol. 14, no. 7, 1996, pp. 1228—1234.

[12] A. Haider, et. al., , “Challenges in Resource Allocation in Network Virtualization”, 20th ITC Specialist Seminar, 18.-20. May 2009, Hoi An, Vietnam, http://www.itcspecialistseminar.com/paper/itcss09Haider.pdf (last access Feb 2013)

[13] E. Borcoci, R. Miruta and S. Obreja, “Multi-domain Virtual Content-Aware Networks Mapping on Network Resources” , 20th European Signal Processing Conference, August 2012, pp. 2223-2227

[14] D. Adami, et. al., Design and Performance Evaluation of Service Overlay Networks Topologies, Journal of Networks, Vol. 6, No. 4, April 2011.

[15] A. Karamoozian, M. Erfani and A. H. Abdullah, "QoS-Satisfied Dynamic Routing based on Overlay Service Network", Second International Conference on Communication Software and Networks, ICCSN 2010, Feb 2010, pp. 441-445

[16] http://atoms.scilab.org/toolboxes/NARVAL (last access March 2013)

[17] http://www.scilab.org/ (last access March 2013)

[18] http://www2.math.uu.se/research/telecom/software/stgraphs.html (last access March 2013)