Online Management of QoS Enabled Overlay Multicast Services

Bart De Vleeschauwer, Filip De Turck, Bart Dhoedt and Piet Demeester
Ghent University - IBBT - IMEC, Department of Information Technology
Gaston Crommenlaan 8 bus 201, 9050 Gent, Belgium
Tel: +3293314900, Fax: +3293314899
Email: bart.devleeschauwer@intec.ugent.be

Abstract— **More and more, content providers offer multimedia services such as Internet TV, multimedia conferencing and online gaming to their customers. These services are characterized by their high sensitivity to network delay and a multicast nature. An overlay network allows for supporting QoS by making reservations in the underlying networks and for multicasting the multimedia streams towards their targets at the overlay layer, without requiring multicast support from the underlying networks.**

This paper outlines the architecture of a dynamic QoS enabled multicast overlay network and also introduces a set of algorithms to determine an overlay distribution tree that connects a multimedia server to a number of clients. The algorithms construct a tree with a bounded end-to-end delay and minimize the bandwidth that is used. These algorithms are evaluated in terms of bandwidth cost, overlay cost and end-to-end delay. We show that one of our heuristics finds overlay multicast trees that approximate the optimal result in terms of cost and that have a small diameter and a low average delay.

I. Introduction

Service providers offer a myriad of multimedia services such as online gaming, multimedia conferencing and Internet TV to large numbers of distributed clients. These services share some common characteristics. On one hand, network delay and loss have a huge impact on the Quality of Experience (QoE) of the users for all these services, rendering them very sensitive to congestion. Therefore, multimedia content providers need to reserve significant amounts of bandwidth to be able to have guaranteed network Quality of Service (QoS). On the other hand, they all need to send multimedia streams from one or more central points to a whole range of clients. It is obvious that these services would greatly benefit from a ubiquitous implementation of IP layer multicasting. However, network layer multicasting is only supported scarcely in the Internet. One of the reasons for this is that it requires all the routers to maintain state information for the multicast groups that use them. As a result, multimedia content providers that want to offer guaranteed service to their customers need to make significant investments to reserve bandwidth along the unicast paths in the network, to offer quality guarantees to all their clients.

Service Overlay Networks (SONs) offer a solution to this problem. A Service Overlay Network is a network where the nodes are overlay servers and the edges are virtual overlay links between pairs of overlay servers. The connectivity of these virtual links is actually provided by the end-to-end paths between the servers. In the overlay network, traffic can be multicasted, regardless of the presence of multicasting support in the underlying network(s). To set up a multicast session, the overlay provider reserves bandwidth along an overlay multicast tree that connects the source of the multicast connection to its target nodes. The SON gives the multimedia service provider the ability to delegate the actual distribution of the multimedia data. Because the overlay network can multicast the data, it uses only a fraction of the bandwidth that would be consumed if direct unicast connections from the multimedia server to the clients were used. However, when multicasting the multimedia streams at a higher layer, an overlay provider must still meet the QoS requirements of the services and the overlay network must avoid excessive delays on the paths to the end users.

One of the key problems faced by the overlay providers is how to set-up the multicast tree that connects the source of a multimedia stream to its clients. This distribution tree needs to balance two requirements. On one hand, the overlay provider wants to minimize the cost and the bandwidth that is reserved in the network, on the other hand, the delay constraints between the multimedia server and each client may not be violated. To solve this problem we have developed algorithms to build a distribution tree with a bounded delay and a minimal cost. In doing this, the overlay provider is not only able to optimize its cost, but it also bounds the end-to-end delay that the service experiences, thereby implementing a fairness over all the clients and meeting the strict QoS requirements of the services. To construct such a tree, we have developed an optimal algorithm as well as two heuristics. As the problem of finding a bounded delay minimal cost overlay tree is a specific instance of the Minimal Steiner Tree problem, it is also NP complete [1]. Therefore, the optimal algorithm only scales up until a limited number of clients per connection.

This paper is structured as follows: section II contains a description of related work. Section III gives an architectural overview of the overlay network we consider in this paper. Section IV discusses a formal problem formulation for constructing a minimal cost tree that bounds the end-to-end delay and a description of the algorithms that were developed. In section V, the evaluation of the algorithms is presented. Conclusions are drawn in section VI.

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

II. Related Work

Overlay networks are able to enhance the standard network services in various domains. In [2], the authors study the impact of overlay server location and path selection on the overall overlay acceptance rate. In [3] a QoS enabled unicast routing algorithm for overlay networks is proposed. Overlay networks are also used as a means to offer extra resilience in [4] and [5] studies the impact of the overlay topology on the resilience of overlay networks. In [6], [7], the concept of overlay layer multicasting is thoroughly discussed. In [8], an algorithm for determining a Single Source overlay Multicast tree is also given and in [9] an algorithm for determining the overlay distribution tree for conferencing applications is described.

The algorithms presented in this paper take special care to minimize the number of overlay servers present in the tree.

III. Architecture Description

This section contains a description of the architecture of the SON. The SON takes care of all the network aspects for multimedia services. A multimedia content provider only needs to contact the overlay broker, which takes care of all the other aspects of the actual delivery of the multimedia streams. This allows the content provider to focus on its service and provides a separation between the network and service aspects of the multimedia service. In doing so, the SON forms a layer between the actual content providers and the network itself. There are a number of components in the architecture that are related to the multimedia service, the overlay service and the network service:

1) Bandwidth Broker (BB): This network entity allows to make on-demand bandwidth reservations. When a service needs an amount of bandwidth between two points in the network, it contacts the BB. The BB knows the current state of the network bandwidth and makes a reservation of the required bandwith between the end points in the network.

2) Overlay Broker (OB): The overlay broker is a management plane component that processes request from the multimedia services. The overlay broker is responsible for computing an ideal distribution tree for a multimedia session. To compute the distribution tree, the OB uses the overlay topology. The overlay topology contains on the one hand the overlay servers and on the other hand, the virtual links connecting them and information on the cost of reserving bandwidth on a virtual link and on the end-to-end delay that is experienced on this link. We assume that the OB has full knowledge on the delay between two OSs and the cost to reserve bandwidth between them. This information could be obtained through interaction with the BBs or by monitoring the virtual links. After a tree has been determined, the OB contacts the BB of the domains that the multicast tree uses and reserves the required bandwidth between the participating Overlay Servers in these domains. In a last phase, the OB also configures the overlay servers to forward the multimedia stream along the tree towards the clients. The OB makes sure that the QoS requirements are met when reserving the bandwidth for a service and when setting up the distribution trees.

3) Overlay Server (OS): The overlay servers are the network nodes of the overlay network. They are configured by the OB and forward multimedia streams to the actual clients and to other overlay servers.

4) Multimedia Server (MS): The multimedia servers offer a multimedia service to their clients, this can be an Internet TV service or a multimedia conferencing service. The MS contacts the OB to setup a distribution tree to deliver its service to all its clients. These multimedia services require an amount of bandwidth and also have restrictions on the end-to-end delay they may experience.

5) Multimedia Client (MC): A multimedia client is a client that receives a multimedia service. These services are offered by the MS and delivered via the SON.

Fig. 1 illustrates how a connection is set-up for a multimedia service and how the stream is forwarded at both the overlay and the network layer.

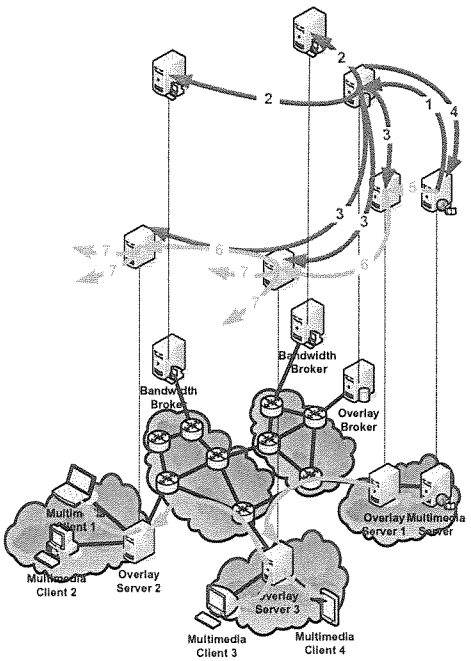


Fig. 1. Using an overlay network to send a multimedia stream to a number of clients. First the multimedia server requests the connection via the OB (1). Then the OB makes the necessary reservations with the Bandwidth Brokers (2) and configures the Overlay Servers to forward the traffic correctly (3). It then notifies the multimedia server that all the necessary reservations are made (4). After the connection has been set-up, the multimedia server forwards the stream to the overlay server in its domain (5), this overlay server and the other intermediate overlay servers forward the stream along the multicast tree (6) and towards the clients of the connection (7). The bandwidth brokers in the client networks are not shown.

The remainder of this paper focuses on algorithms that can be used by the OB to compute the overlay distribution tree.

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

IV. Bounded Overlay Tree Problem

The bounded minimal cost overlay tree problem is formulated as follows: Given an overlay graph $G = (V, E)$ with node set V and edge set E. In the overlay graph, two weight functions are present, $C(e)$ and $D(e)$, for the edges of E. These functions represent the cost to make a reservation on an edge and the end-to-end delay that is experienced on an overlay edge. The values of $C(e)$ and $D(e)$ are positive integers for all the edges of E and the values of $C(e)$ and $D(e)$ are determined by the cost and the delay on the end-to-end Internet path that connects the endpoints of e. Given a multicast connection c, which consists of a source overlay server s, a set of target overlay servers $T = \{t_1 \dots t_n\}$ and a delay bound D, a bounded overlay tree Tr is a tree in G, rooted at s, that spans the nodes in T in such a way that for each node t_i in T, the delay on the path from s to t_i in the tree is at most D. A bounded minimal cost overlay tree is a bounded cost overlay tree such that $\sum_{e \in Tr} C(e)$ is minimized. An additional optimization criterion is that we want a tree where the number of overlay edges that are used is also minimized. Since an overlay network has a full mesh topology, several trees with the same overall cost but a different number of intermediate overlay servers can be available. Because the overlay servers consume access bandwidth and processing power, we look for the bounded minimimal cost overlay tree with the least number of overlay servers. This tree is a minimal cost overlay tree that minimizes $|Tr|$. The problem that is faced by the overlay broker is finding such a tree with a low cost in a timely fashion.

A. ILP based algorithm

To solve the bounded minimal cost overlay tree problem, an Integer Linear Programming (ILP) [10] formulation was developed. The cost function in the formulation minimizes the cost of the IP links that are used to establish a multilayer multicast connection and the number of overlay servers involved for a minimal IP cost. Solving the formulation returns the optimal solution to the problem. However, solving ILP problem formulations does not scale well, making it only suitable for overlay networks with a limited number of overlay servers and only a few target nodes for a connection. For more details on this formulation, the reader is referred to [2].

B. Subnetwork Heuristic Algorithm

The subnetwork heuristic algorithm (SUB) is based on the observation that for every target node, only a small part of the full overlay topology can be used to send the stream from the source node to this target node, without violating the delay constraint. Therefore, we find the Subnetwork for every target node that contains low delay paths from the source to that target node. The algorithm then searches through these subnetworks and builds the actual multicast tree that will be used for all the target nodes. In the consecutive iterations, care is taken that the delay bound is never violated. After constructing an initial multicast tree, an optimization algorithm is applied to this tree, to further reduce the overall cost in terms of IP bandwidth and used overlay servers.

1) Determining the subnetwork: To find the subnetwork $S_i = (V_i, E_i)$ of G that can be used to send data from a node s to a node t_i without violating the delay constraint D, we use a function $D_{min}(v_1, v_2)$, that returns the minimal delay in the overlay graph between nodes v_1 and v_2 in V. This structure is maintained by the OB, using the overlay topology and the measurements in the overlay network. To further optimize the end-to-end delay that is experienced in the multicast tree, the delay bound for determining S_i was made dependent on the minimal overlay delay between s and t_i. For t_i, the delay bound D_i is set to $min(\alpha \times D_{min}(s, t_i), D)$, α is a parameter of the algorithm that determines how many nodes are present in the subnetwork and also has an impact on the delay that will be experienced for a specific target node in a tree. The nodes of S_i are nodes for which:

$$\forall v \in V_i : D_{min}(s, v) + D_{min}(v, t_i) \leq D_i \quad (1)$$

An edge (v_k, v_l) belongs to the subnetwork if:

$$\forall (v_k, v_l) \in E_i : D_{min}(s, v_k) + D((v_k, v_l)) + D_{min}(v_l, t) \leq D_i \quad (2)$$

To determine the nodes and edges of S_i, it suffices to check whether the nodes and edges in G obey the equations 1 and 2. The cost to determine this for one $s - t_i$ pair has a complexity of $O(m^2)$, with m the number of nodes in the overlay network.

2) Core algorithm: The algorithm finds a path from the source node to every target node. For every target node $t_i \in T$, the algorithm starts at the source node. The algorithm then looks for the most suitable node in the subnetwork S_i to be the next hop in the path towards t_i. To select the best node, a decision is made, based on the number of subnetworks the node belongs to, the cost of the overlay edge between the source node and the candidate node and the cost of the path between the candidate node and t_i. We define the presence of a node as the number of subnetworks from remaining target nodes, the candidate node belongs to. The weighted cost of an edge is the cost of the edge between the current node and the candidate node, divided by its presence. We select the node with the lowest weighted cost. If two nodes have the same weighted cost, the node that has the minimal cost to target node t_i is chosen. Once a node v_{opt} is selected, a number of nodes are removed from the subnetwork S_i. The node itself is removed and also the nodes v with a delay towards the target node that exceeds the delay from the selected node to the target node. This is done to make sure that the algorithm finds a path that is closing in on t_i. Nodes that can no longer be used to find a path to t_i without violating D are also removed from S_i. The process of selecting a node is repeated until finally the target node is reached. The algorithm then proceeds with the next target node and sets the cost of the intermediate edges of the path for the previous target node to 0. By doing this, the algorithm gives an advantage to edges that are already used to reach other target nodes.

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

```
Input : Network N
        Source s,
        Delay Bound D,
        Target Set T = {t_1 ... t_n}
        SubNetwork Set S = {S_1 ... S_n}
Output: Tree
Algorithm:

01 Tar=T.first
02 while(!T.isEmpty())
03  SubNet=S.get(Tar)
04  Current=Source
05  TotalDelay=0
06  Path=s
07  while(!(Current==Target))
08   BestNode = null
09   BestEdge = null
10   for(Node ∈ SubNet)
11    if(Test(Node, BestNode))
12     BestNode=Node
13     BestEdge=(Current, Node)
14   Path=Path+BestEdge
15   TotalDelay+=D(BestEdge)
16   SubNet=Reduce(SubNet, TotalDelay,
                   D_min, BestNode)
17   if(BestNode ∈T)
18    T.remove(BestNode)
19   Reorder(T, BestNode, TotalDelay)
20   Current=BestNode
21  ChangeCosts(Path, N)
22  Tree=Tree+Path
23  T.remove(Tar)
24  Tar=T.first
25 Optimize(Tree)
26 return(Tree)
```

Fig. 2. The pseudo code of the algorithm, the algorithm is given an overlay network N, a source node s, a delay bound D, a set of target nodes T and a set of relevant subnetworks T and computes a bounded overlay multicast tree.

The algorithm is presented in pseudo code in fig. 2. It starts by selecting the first node of the target set and looks for a path to this node. The "Current" variable contains the progress towards the target node. It starts with the source node s and is replaced as the algorithm is getting closer to the actual target node. The algorithm evaluates all the nodes that are in the subnetwork (line 10-13), this is done by comparing the node with the best candidate that was found up until this moment. If it is better, it replaces the BestNode. Once all the nodes have been evaluated, a new edge is added to the path for the target node, the value for the total delay is adjusted and the subnetwork is adapted (lines 14-16). The subnetwork is changed by eliminating nodes that are no longer valid for the source-target path, for instance because it is no longer possible to find a path that goes through this node without violating the delay bound. Additionally, we also require that a node is always closer in terms of delay to the target node than the previous node in the path, this is enforced by removing nodes that do not have this property from the subnetwork. A check is also performed on whether the selected node was another target node (lines 17-18). If this is the case, it is removed from the target set. Additionally, the algorithm also reorders the target set. In this step, the target nodes that have subnetworks that contain the best node that is chosen are put in the first positions of the T set. By doing this, the next target node that will be processed, will share a common path with the target node that we are processing now. This node selection process is repeated until the target node is reached. The values for the cost of all the edges in the path that have been used to reach this target node are set to 0 (line 21) and a path for a new target node is searched. This is repeated while the target set is not empty. After all the target nodes have been processed and paths have been found to each one of them, the optimization phase is started and finally, the optimized tree is returned. To eliminate the influence of the first target node on the solution of the problem, the algorithm can be run separately, starting with every target node. The algorithm is of polynomial complexity in the number of overlay servers and target nodes and finishes much faster then the ILP based algorithm. In fig. 4 an example of the core Subnetwork algorithm is illustrated. This example does not include the optimization phase and assumes that the subnetworks have already been calculated, they are illustrated with dashed lines. The underlying network topology is not shown in this figure.

3) Optimization Phase: The core algorithm determines the paths followed for all the connections. However, there still might be some redundant overlay servers or overlay links in the paths. Therefore, the algorithm is followed by an optimization phase, where redundant overlay links or overlay servers are removed from the paths.

- Eliminating double paths: It is possible that the algorithm finds two paths going through the same intermediate server when trying to find the paths for two target nodes. When this occurs, the path to these target nodes is replaced by the path with the lowest delay. This is illustrated in fig. 3.

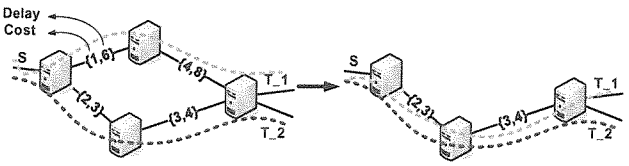


Fig. 3. Eliminating redundant overlay links and server from a path, due to an overlap between the paths to two target nodes.

- Eliminating redundant overlay servers: When the algorithm finds a path to a target node, it is possible that an overlay server is present redundantly in the path. This might happen when the direct connection between two overlay servers in the path, with one or more intermediate overlay server does not cost more than the path via the overlay server and does not violate the delay constraint. These intermediate overlay servers can be removed from the path and they can be replaced by a direct connection between the two other servers. This example is illustrated in fig. 5.

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

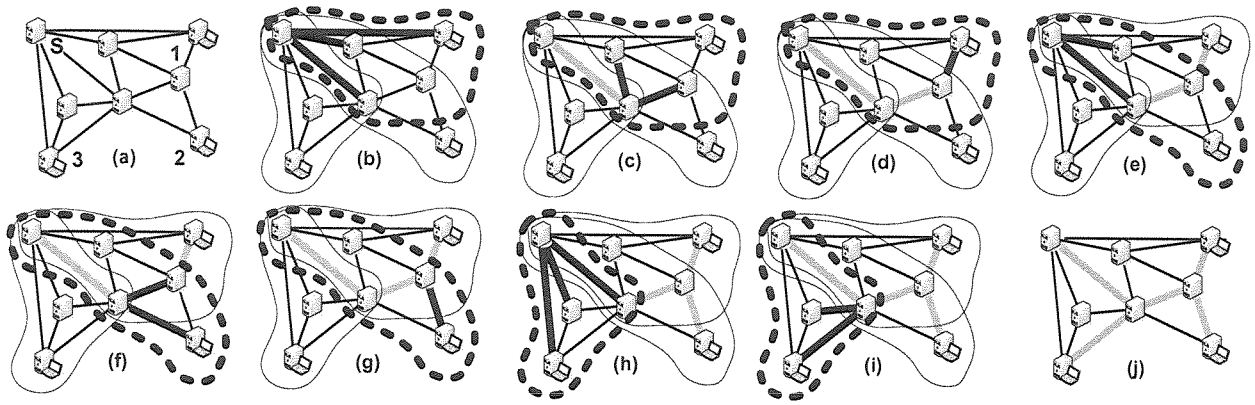


Fig. 4. This example illustrates the core phase of the algorithm: a connection is set up between a source server S and 3 target servers located in client domains. The subnetworks for the clients are illustrated with a dashed line. Both cost and delay are supposed to be equivalent with the distance in the figure. The algorithm starts by finding a path to client 1, 3 links are possible and the link that is present in the most subnetworks is chosen. In the next step, of the two possible links, the one that is closest to both clients 1 and 2 is chosen and the last edge finally reaches client 1. The first two edges for client 2 have cost 0, since they are already present in the path for client 1, the last edge reaches client 2. For client 3, an analogous decision process is followed.

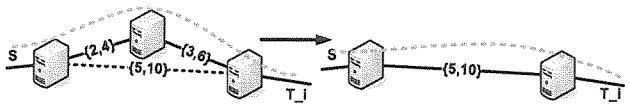


Fig. 5. Eliminating an overlay server from the path, due to a better direct link.

C. Bounded Spanning Tree Algorithm

This heuristic is a variation on the standard minimal spanning tree algorithm [11]. It only includes nodes that are also target nodes of the connection. It starts with the source node and adds the overlay link to the target node with the least cost to the tree that does not violate the delay constraint that is required from the tree. It then proceeds to add edges to the tree that are connected to either the source node or a target node that was already reached, that has the least cost and that does not violate the delay constraint. One of the advantages of this algorithm is its low complexity. Another property is that the number of overlay servers that are involved in the connection is bounded by the number of target nodes.

V. Evaluation of the Algorithms

We used simulations to evaluate the behavior of the algorithms described in the previous section. The topology of the network that we used is shown in fig. 6 and has 37 nodes and 114 unidirectional edges. To generate an overlay network, nodes were chosen at random in the network and connected in a full mesh virtual topology. The cost to reserve bandwidth for an overlay edge was proportional to the number of network edges on the network path that connects the endpoints. The delay in the network was proportional to the actual physical distance. We assumed that a shortest hop count path routing policy was used by the network provider. The results that are presented in this section are averaged over 1000 iterations, where different overlay servers were selected and the source node and target nodes of the multicast connections were varied. To obtain a multimedia connection, a source node and a number of target nodes are chosen randomly from the network, all the nodes had an equal probability to be chosen. The value for the α parameter of the sub algorithm was 2. The delay bound for a connection was set to the worst of the end-to-end unicast delays between the source node and the target nodes. The notations IP, ILP, SUB and BST are used to denote the behavior of using standard network unicast routing and reservations and using the overlay multicast trees generated with the ILP based algorithm, the Subnetwork heuristic and the Bounded Spanning Tree heuristic.

A. Network Cost

One of the most important parameters from the viewpoint of an overlay provider is the actual cost to support a multimedia connection. In this section we present the results for the cost to reserve bandwidth to support multimedia connections with a varying number of target nodes. Fig. 6 shows the cost to reserve bandwidth for a multimedia connection with an increasing number of clients in an overlay network of 20 servers. The graph illustrates the gain that is achieved by using overlay layer multicasting, especially with a large client set, the cost for the overlay multicast routing algorithms are much lower than that of standard IP routing. The graph also shows that all the algorithms for building an overlay multicast tree decrease the cost considerably. In fig. 7 the cost difference between the heuristics and the optimal solution (the result of the ILP algorithm) is shown. Here we see that the BST heuristic is outperformed by the SUB algorithm. On average, the SUB algorithm stays within 7% of the optimal solution, even with 12 clients for a multimedia connection. The cost in terms of number of overlay edges is also shown, one can observe that the ILP and SUB algorithm use more overlay edges to deliver their services than the BST algorithm. This is explained by the fact that the BST algorithm uses only target nodes to build the tree and by the usage of intermediate nodes by the other algorithms.

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

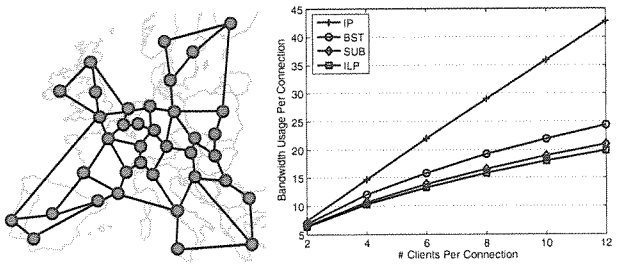


Fig. 6. The network topology on which the algorithms were evaluated and a graph showing the cost of reserving bandwidth in the network for the standard unicast IP case and overlay multicast trees constructed with the optimal algorithm (ILP), the subnetwork heuristic (SUB) and the bounded minimal spanning tree algorithm (BST)

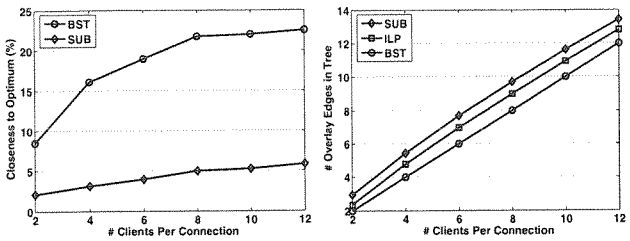


Fig. 7. The relative difference in bandwidth cost between the solution provided with the optimal algorithm and the two heuristics for an increasing number of target nodes. The second graph shows the number of overlay edges used in the trees for the three algorithms.

B. End-to-End Delay

The average and worst delay that are experienced by the multimedia clients are essential to have a good overlay routing service. In fig. 8, the relative difference between the average and worst delay for the trees returned by the different algorithms and when using IP unicast routing are shown for an overlay network of 20 overlay servers and for connections with one source server and 10 clients. In the second graph, 10 network edges were chosen at random and were given a much higher delay value. This was done to evaluate the ability of the algorithms to provide low delay routes when some links exhibit excessive delays. For the normal scenario, one can see that the worst delay is almost the same for all the algorithms. The average delay is slightly higher for the overlay approach. The reason for this is that the overlay must balance cost and delay, to decrease the overall cost, the average delay will increase slightly, however, as the delay bound is not violated, the QoS of the connection is maintained. The SUB heuristic has an average delay that is closest to the IP delay. In the congested scenario, we observe that the delay of the SUB and ILP algorithms are much lower than that with standard IP routing. This is due to the overlay network ability to route around parts of the network with a high delay at the overlay layer. In this way, the overlay network is able to offer a much better QoS than that with shortest hop count path at the network layer. The BST algorithm is not able to match this low average delay as it is restricted to target servers and only takes the reservation cost and the worst delay bound into account.

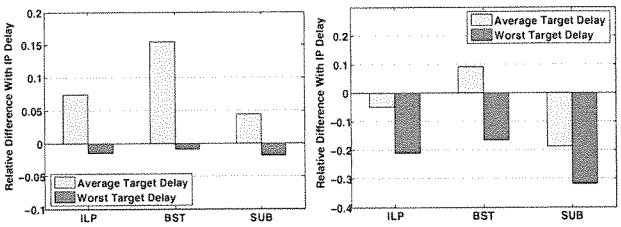


Fig. 8. The relative difference between the average and worst delay that is experienced for connections with 10 target nodes in a 20 server overlay network for the trees returned by the algorithms and that with standard IP unicast routing. The results for two scenarios are depicted, graph one has no congestion, graph two is for a network with 10 congested network edges.

VI. Conclusions

In this paper we introduced an architecture of a QoS enabled multicast overlay network built to support services such as Internet TV and online gaming. A number of algorithms were presented to construct an overlay multicast tree that achieves decent QoS for all the target nodes and that has a low cost in terms of used network bandwidth. These algorithms were evaluated in terms of used bandwidth, number of overlay servers in the tree and the average and worst end-to-end delay. We found that it is possible to calculate a tree with a minimal cost and a bounded delay in a scalable way. For a 20 node overlay network and multicast connections of up to 12 targets, the cost was within 7 % of the optimal bounded minmal cost tree for the Subnetwork Heuristic.

References

[1] R. E. Miller and J. W. Thatcher, Eds., *Complexity of computer computations*. Plenum Press, 1972.

[2] B. De Vleeschauwer, F. De Turck, B. Dhoedt, and P. Demeester, "On the construction of QoS enabled overlay networks," in *Quality of Future Internet Services (QofIS)*, ser. Lecture Notes in Computer Science, vol. 3266, september 2004, pp. 164–173.

[3] Z. Li and P. Mohapatra, "QRON: QoS-aware routing in overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 29 – 40, 2004.

[4] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Symposium on Operating Systems Principles*, 2001, pp. 131–145.

[5] B. De Vleeschauwer, F. De Turck, B. Dhoedt, and P. Demeester, "Dynamic algorithms to provide a robust and scalable overlay routing service," in *The International Conference on Information Networking (ICOIN)*, ser. Lecture Notes in Computer Science, vol. 3961, 2006.

[6] Y. Chu, S. Rao, and H. Zhang, "A case for end system multicast," in *SIGMETRICS '00: Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, 2000, pp. 1–12.

[7] L. Lao, J.-H. Cui, and M. Gerla, "Multicast service overlay design," in *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'05)*, 2005.

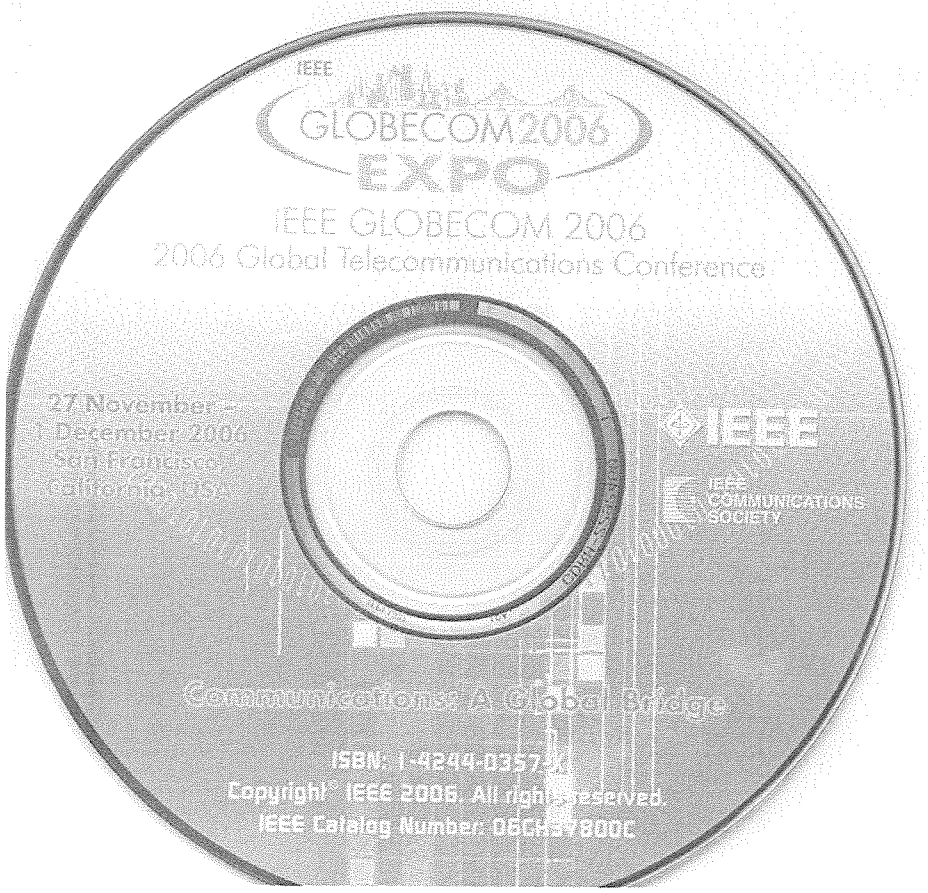
[8] X. Zhang and G. Zhang, "A multicast routing algorithm for overlay network built on leased lines," in *SAINT '05: Proceedings of the The 2005 Symposium on Applications and the Internet (SAINT'05)*, 2005, pp. 118–124.

[9] S. Aggarwal, M. Limaye, A. Netravali, and K. Sabnani, "Constrained diameter steiner trees for multicast conferences in overlay networks," in *QSHINE '04: Proceedings of the First International Conference on Quality of Service in Heterogeneous Wired /Wireless Networks (QSHINE'04)*, 2004, pp. 262–271.

[10] L. A. Wolsey and G. L. Nemhauser, *Integer and Combinatorial Optimization*. Wiley-Interscience, 1988, vol. 0-471-82819-X.

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition*. The MIT Press, September 2001.

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.


IEEE
GLOBECOM2006
EXPO
IEEE GLOBECOM 2006
2006 Global Telecommunications Conference
27 November –
1 December 2006
San Francisco
California USA
IEEE
IEEE COMMUNICATIONS SOCIETY
Communications: A Global Bridge
ISBN: 1-4244-0357-X
Copyright © IEEE 2006. All rights reserved.
IEEE Catalog Number: 06CH37800C