HeteCast: A Heterogeneous Qos Overlay Multicast Routing Scheme

Jun Lu Qiuqi Ruan Rongrong Ni

Institute of Information Science
Beijing Jiaotong University
Beijing, 100044, China
lujuncn@hotmail.com, qqruan@center.njtu.edu.cn, nirongrong@263.net

Abstract

This paper presents a heterogeneous Qos overlay multicast routing scheme (*HeteCast*) for Qos-based heterogeneity and scalability issues in multicast. The scheme seeks to implement differentiated Qos support within one multicast tree and balance related tree cost metrics for multicast session scalability. Besides efficient metric tradeoff routing performance based on our previous work, *HeteCast* extends heterogeneous Qos support in the path selection and maintenance. Simulation experiments demonstrate the scheme achieves not only better heterogeneous Qos data forward control, but also better routing scalability by efficient routing metrics tradeoff.

Keywords: overlay multicast, routing, heterogeneity

1. Introduction

Cooperative applications based on group communication are expected to become widespread in Next Generation Network[1]. And multicast is an efficient transmission for group communication thanks to its capability for bandwidth conservation[1]. However, IP Multicast wide deployment seems far from expectation because of many reasons. Among these, one is multicast-enable routers deployment problem, the other is heterogeneity and scalability issues of multicast itself.

Researchers have provided two key technologies to the two problems above respectively. One is Overlay multicast[2], and the other is DiffServ[4]:

Overlay multicast is an application-level multicast based on a set of distributed Multicast Service Nodes (MSN: a kind of application server directly accessing to near access router with high access bandwidth) that provides multicast services to end users on top of the general Internet unicast infrastructure[2]. As it does not require changes to the existed Internet Infrastructure, overlay multicast can be easily deployed [3].

DiffServ was proposed to provide aggregated Qos support in the Internet for application scalability[4]. Multicast heterogeneity includes terminator heterogeneity, network heterogeneity, and Qos heterogeneity. Based on the basic idea of DiffServ, some solutions[5] also have proved to be efficient in the heterogeneity and scalability issues.

Overlay multicast or DiffServ is effective to its own issue. When combining these two together, we need to consider a new challenge: how to deliver DiffServ (Heterogeneous Qos) in an overlay multicast tree and achieve scalability.

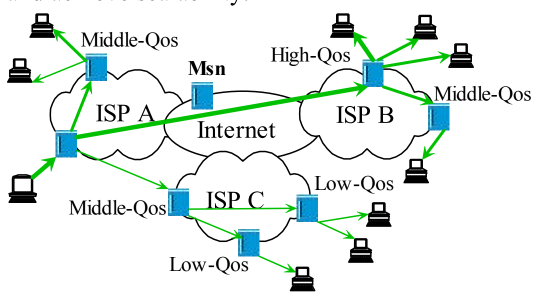


Fig. 1 Heterogeneous Qos in Overlay Multicast

This paper proposes a heterogeneous Qos overlay multicast routing scheme (*HeteCast*) to the challenge above. Besides the effective metrics tradeoff routing performance based on our previous work[6], *HeteCast* extends heterogeneous Qos support in the path selection and maintenance by leveraging MSN Qos-level, defining MSN Qos-level link, and constructing differentiated Qos path for data forward. Moreover, by balancing related tree cost metrics, the scheme has better scalability in overlay multicast session size. Therefore, the multicast group can efficiently construct heterogeneous Qos routing tree, and effectively expand to large session size shown in Fig 1.

The rest of the paper is structured as follows. Section 2 describes the overview and. Section 3 describes *HeteCast* Heuristic scheme. In Section 4 we evaluate its performance and finally draw a conclusion in Section 5.

This work is supported by Beijing Municipal fund SYS100040409.

2. HeteCast:Overview

2.1. Basic Idea of HeteCast

HeteCast aims at implementing a single tree to support heterogeneous Qos in overlay multicast. Based on the idea of DiffServ[4] and our previous work on overlay multicast routing[6], the scheme first divides MSNs into different Qos-level groups (High-Qos, Middle-Qos, and Low-Qos) for heterogeneous service, and leverages MSN access bandwidth according to Qos level of its related overlay path link. Secondly the scheme appends each MSN level by level to the tree in the course of the initial tree construction, in which higher Qos MSN will have shorter overlay delay and better service. Thirdly the scheme will optimize the routing tree by carefully considering routing metrics and Qos effect in tree maintenance by means of heuristic local search.

Figure 1 above demonstrates the idea of our scheme. The root MSN is responsible to construct the initial tree, and all MSNs will work together to maintain and optimize the tree topology. The tree will adapt itself to support not only efficient heterogeneous Qos overlay path selecting, but also effective MSN access bandwidth usage ratio saving.

2.2. Heterogeneous Qos Extending

HeteCast scheme is based on our previous work Cost-Balanced Overly multicast routing scheme (*Cost-Balanced*) [6]. Its basic idea is to create a directed spanning tree rooted at the source MSN to tradeoff two routing metrics overlay path delay and MSN access bandwidth usage ratio by carefully selecting a 'less-MSN-access-bandwidth' path within minimum -delay overlay paths to lessen MSN access bandwidth as much as possible by delay threshold control.

HeteCast directly extends *Cost-Balanced* by leveraging Qos in MSNs to the heterogeneity issue in the overlay multicast. As a result, *HeteCast* is a routing metrics optimization problem as *Cost-Balanced*. We select local search to optimize in our scheme, and take the term aggregate subtree overlay latency Λ_i [3] and similarly define a term called aggregate subtree access bandwidth Δ_i (considering MSN heterogeneous Qos link)as the local search optimization criteria for MSN i.

S_i denotes the entire set of clients served by all MSN in the subtree rooted at i. s_i is the number of S_i given by:

$$s_i = |S_i| = c_i + \sum_{j \in children(i)} s_j \quad (3\text{-}1)$$

In the equation (3-1), $children(i)$ indicates the children set of i on the overlay tree. c_i denotes the number of clients directly served by MSN i.

Λ_i denotes the summation of overlay path delay of each MSN in the subtree rooted at MSN i, weighted by the number of clients served directly or indirectly by MSN i.

$$\Lambda_i = \sum_{j \in children(i)} (s_j l_{i,j} + \Lambda_j) \quad (3\text{-}2)$$

In the equation (3-2), $l_{i,j}$ is the unicast latency between MSN i and its children MSN j.

Δ_i denotes the summation of MSN access bandwidth usage ratio of each MSN in the subtree rooted at MSN i, weighted by the number of clients served directly or indirectly by MSN i.

$$\Delta_i = \sum_{j \in children(i)} (s_j r_{i,j} + \Delta_j) \quad (3\text{-}3)$$

In the equation (3-3), $r_{i,j}$ is MSN access bandwidth usage ratio at MSN i used by its children MSN j. We utilize MSN degree[1] to represent $r_{i,j}$: suppose d_i is total degree at MSN i, then different Qos-level child MSN j linking to MSN i has different access bandwidth usage ratio $r_{i,j} = d_{i,j} / d_i$, in which $d_{i,j}$ are 1, 0.75, 0.50 for different Qos level link to MSN i. (Suppose High-Qos-link access bandwidth degree is one unit, Middle-Qos-link and Low-Qos-link related MSN degree consumption are 0.75 and 0.50 respectively.)

The optimization objective of *HeteCast* is to minimize the average of the aggregate subtree overlay latency at root MSN $\overline{\Lambda_r} = \Lambda_r / s_r$ while to select the optimal path to minimize the average of the aggregate subtree access bandwidth at root MSN $\overline{\Delta_r} = \Delta_r / s_r$ as much as possible. Here $\overline{\Lambda_r}$ stands for how much total average overlay path delay used by a client in the routing tree, and $\overline{\Delta_r}$ illuminates how much total average access bandwidth usage ratio provided by all MSNs to a client in the tree.

For brevity, in the rest of this paper, we will refer overlay path delay to overlay delay, MSN access

[1] Assuming source MSN traffic data packet at rate of B bps, If the total access bandwidth at MSN i is b_i bps, the MSN degree at i is $d_i = [b_i / B]$. That is, the MSN i can create at most d_i unicast links to other MSNs[2].

bandwidth usage ratio to access bandwidth, and aggregate subtree overlay latency to aggregate latency, and aggregate subtree access bandwidth to aggregate bandwidth respectively.

3. HeteCast Heuristic Scheme

The scheme is a distributed solution and includes two parts: initial tree construction and successive tree optimization.

3.1. Initial Tree Construction

The task is to organize the initial join MSNs into a routing tree for the following local search optimization. All kinds of degree-restraint SPT-based algorithms are available. Here we directly apply the basic idea of the initial construction algorithm in the *min-avg-latency* scheme[3] to our scheme for the following performance comparison convenience, only considering MSN Qos level priority in the construction.

3.2. Tree Successive Optimization

We carry out local transformations to fulfill the local search objective above based on our previous work [6], which extends to consider the differentiated Qos in these operations including two steps. The first step is local transformation operation and the second step is its heterogeneous Qos support.

(1) Local transformation operation

Local Transform (g , id)

1. $\overline{\Lambda}_{\min} = \overline{\Lambda}_g$; $\overline{\Delta}_{\min} = \overline{\Delta}_g$; *availID* = -1
2. Get available set M for No. id type local transformation
3. For all $i \in M$
4. Calculate $\overline{\Lambda^i}_g$ and $\overline{\Delta^i}_g$ for available operate i
5. If ($\overline{\Lambda^i}_g < T + \overline{\Lambda}_{\min}$ and $\overline{\Delta^i}_g < \overline{\Delta}_{\min}$ and) then
6. $\overline{\Lambda}_{\min} = \overline{\Lambda^i}_g$ and $\overline{\Delta}_{\min} = \overline{\Delta^i}_g \rightarrow$ *availID*= i
7. End-if 5~7 --- ***HeteCast***

5'. If ($\overline{\Lambda^i}_g < \overline{\Lambda}_{\min}$) then -

6'. $\overline{\Lambda}_{\min} = \overline{\Lambda^i}_g \rightarrow$ *availID* = i

7'. End-if 5'~7' --- ***min-avg-latency***

8. End-for
9. MSN g No. id local transform operation by *vailID*

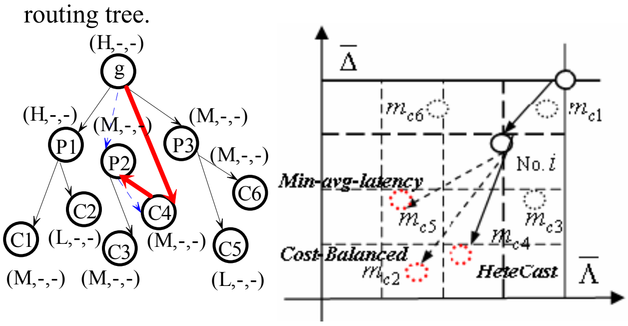
Fig.2 Local Transformation for *HeteCast* Routing

In Fig.2, $\overline{\Lambda}_g$ and $\overline{\Delta}_g$ are average aggregate latency and bandwidth of the current topology at MSN g, while $\overline{\Lambda^i}_g$ and $\overline{\Delta^i}_g$ are those of the possible topology i for a certain local transformation. T is the threshold to balance the two routing cost: overlay delay and access bandwidth usage ratio.

(2) Heterogeneous Qos support

After one local transformation available topology satisfies the routing optimization requirements discussed above, it also need to pass the heterogeneous Qos support check. That is, the new Parent MSN Qos-level should Higher than or equal to the new child MSN to guarantee the heterogeneous Qos data forward.

Suppose the current operation is parent-child swap, the check course can be illustrated in Fig.3. To simplify the explanation, MSN's triple only list its *Qos* information ignoring *delay* and *degree*. When MSN g finished No. i local transformation, it will try to find next optimal one. *min-avg-latency* only considers aggregate latency $\overline{\Lambda}$ and then will choose m_{c5} topology as next local transformation (No. $i+1$), while our previous work *Cost-Balanced*[6] will think of both aggregate latency $\overline{\Lambda}$ and aggregate access bandwidth $\overline{\Delta}$ and then select m_{c2} .Different from these two schemes, *HeteCast* will check its related MSN Qos level to support extending function of heterogeneous Qos data forward. As a result, it will choose m_{c4} .As a result, *HeteCast* will effectively save MSN access bandwidth as much as possible, and efficiently implement heterogeneous Qos in one routing tree.



(3-1) parent-child swap (3-2) routing metrics cost

Fig.3 Heterogeneous Qos Support

4. Simulation Evaluation

We designed and developed an event-driven packet-level model based on ns-2 [7] to evaluate our scheme. In the experiments we compared the performance of *HeteCast* with a variation of minimum average latency degree-bounded (*min-avg-latency*) scheme in OMNI [3], which only omitted random-swap local transformation for simplifying the problem and is also shown in Fig.2.

4.1. Routing performance

We first present routing performance in the course of tree successive optimization.

100-node	Delay $\overline{\Lambda}_r$ (ms)			Bandwidth $\overline{\Delta}_r$ (%)		
MSN Size	32	64	96	32	64	96
min-avg-latency	159.06	179.24	213.76	63.29	89.37	106.43
HeteCast	167.78	208.80	213.03	58.13	74.71	101.16

600-node	Delay $\overline{\Lambda}_r$ (ms)			Bandwidth $\overline{\Delta}_r$ (%)		
MSN Size	32	64	96	32	64	96
min-avg-latency	263.23	258.52	312.81	63.89	62.77	126.26
HeteCast	293.64	281.53	352.79	52.31	57.18	103.23

Table 1 Performance for Two Different Size Networks

Table 1 shows the routing performance of our scheme by stable value in the last local transformation of tree aggregate delay $\overline{\Lambda}_r$ (ms) and tree aggregate access bandwidth $\overline{\Delta}_r$ (%). Set 1 (100-node network) and Set 2(600-node network) illustrate that *HeteCast* achieves MSN access bandwidth gain (about 5% to 14%) at a little increase of overlay path latency (about 0.7ms to 40ms), The results implies that the idea of heterogeneous Qos may achieve some gain in MSN access bandwidth for application scalability.

4.2. Scalability

We next analysis *HeteCast* tree cost to show its scalability feature by stable value in the last local transformation of the following statistics variables:

.*Tree total Overlay delay*: the sum of the entire overlay path in the routing spanning tree.

$$L_{overlay} = \sum_{i,j \in T} l_{i,j} \quad i \text{ and } j \text{ are neighbor nodes.}$$

.*Tree contrast MSN access bandwidth*: the sum of all the MSN access bandwidth by means of MSN degree usage ratio in the routing spanning tree.

$$B_{con} = \sum_{i,j \in T} r_{i,j} \quad i \text{ and } j \text{ are neighbor nodes.}$$

Table 2 shows tree cost statistics to demonstrate the scalability feature. The experiments involved 32,64 and 96 MSNs within 100-node network. Tree total Overlay delay $L_{overlay}$ just increases a little, while Tree contrast MSN access bandwidth B_{con} reduces much. In this way, *HeteCast* can expand his service size for better scalability.

100-node	Delay $L_{overlay}$		
MSN Size	32	64	96
min-avg-latency	289.16	518.81	828.93
HeteCast	280.01	576.61	831.71
Ratio①	0.9683	1.1114	1.0033

100-node	Bandwidth B_{con}		
MSN Size	32	64	96
min-avg-latency	7.9907	16.7528	25.7676
HeteCast	6.8338	12.8821	23.2182
Ratio①	0.8552	0.7690	0.9011

①Ratio: HeteCast/min-avg-latency

Table 2 Tree Cost Statistics for Scalability

5. Conclusion

In this paper, we propose a heterogeneous Qos overlay multicast routing scheme (*HeteCast*) for heterogeneity and scalability issues in multicast, which seeks to implement differentiated Qos support within one multicast spanning tree and decrease related cost metrics for scalability. Besides efficient metric tradeoff routing performance based on the previous work, the scheme extends heterogeneous Qos support in the path selection and maintenance by Edge Msn and Qos-level link constructing. The simulation illustrates that *HeteCast* achieves better performance in terms of not only routing metrics tradeoff between overlay delay and MSN access bandwidth, but also better routing scalability by cost metrics compressing. Future work will focus on control overhead compressing for more scalability.

6. References

[1] G. Manimaran, P. Mohapatra. Multicasting: An enabling technology. *IEEE Network*, Vol. 17, Issue: 1, Jan/Feb 2003.

[2] S. Shi. And J.Turner. Multicast Routing and Bandwidth Dimensioning in Overlay Networks. *IEEE Journal on Selected Areas in communications*, Vol. 20, No.8, October 2002.

[3] B. Suman, K. Christopher, K. Koushik, B. Samrat, K. Samir. Construction of an Efficient Overlay Multicast Infrastructure for Real-time Applications. *IEEE INFOCOM*, San Francisco, June 2003.

[4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. weiss. An Architecture for Differentiated Services. *IETF RFC 2475*. December 1998.

[5] A. Striegel, Scalable approaches for DiffServ multicasting, PH.D. Dissertation, Computer Engineering, Iowa State University, 2002.

[6] Jun Lu, Qiuqi Ruan et al. Cost-Balance Overlay Multicast Routing Using Dynamic Estimation. *IEEE TENCON'04*, Chiang Mai, Thailand, 21-24 Nov. 2004.

[7] ns-2 simulator: http://www.isi.edu/nsnam/ns