Performance Evaluation of Multicast Cost Sharing Mechanisms

Nandan Garg and Daniel Grosu
Dept. of Computer Science, Wayne State University
5143 Cass Avenue, Detroit, MI 48202
nandang@wayne.edu, dgrosu@cs.wayne.edu

Abstract

In this paper we investigate experimentally the performance of Marginal Cost (MC) and Shapley Value (SH) mechanisms for sharing the cost of multicast transmissions. We implement and deploy the MC and SH mechanisms on PlanetLab and study their properties. We compare the execution time of MC and SH mechanisms for the Tamper-Proof and Autonomous Node models. We also study the convergence and scalability of the mechanisms by varying the number of nodes and the number of users per node. We show that the MC mechanisms generate a smaller revenue compared to the SH mechanisms and thus they are not favorable for the content provider. From the computational point of view as well as economic considerations, increasing the number of users per node is beneficial for the system implementing these mechanisms.

1. Introduction

Recently, the transmission of multimedia content has become one of the most widely used applications on the Internet. A very common model of multimedia content distribution is the one in which the content provider distributes the audio/video to the subscribers, (*e.g.*, on-demand video) using multicast transmissions. Multicast transmissions are efficient as they minimize the number of messages traversing a single link by creating a multicast tree. The content provider is the root of the multicast tree and the receivers are the nodes in the tree. A node in the multicast tree receives the transmission from the parent node and forwards the content to its children, if any. A multicast transmission incurs a cost (each link in the multicast tree has an associated cost) and it is justifiable that the users who receive the transmission pay for it. In most applications the content provider charges the receivers for providing the content. Thus one important problem needs to be solved, that is, how should the users share the cost of the multicast transmission in a fair way. To solve this problem we need to provide mechanisms for computing the cost shares for each user. Feigenbaum *et al.* [7] used mechanism design theory (a subfield of micro-economics) to design mechanisms for computing the cost shares.

The benefit an user e derives from receiving the multicast is quantified by a private single valued parameter u_e called the user's *utility*. If the cost of the transmission is shared by several users, the users' cost shares will be different (depending on their positions in the multicast tree). If user e's cost share is denoted by x_e, she will like to receive the transmission if her cost share x_e is less than her utility u_e, *i.e.*, if her welfare $w_e = u_e - x_e$ is positive. Cost sharing mechanisms determine who receives the multicast and how much the users have to pay for the service. Such mechanisms ask the users to report their utilities and based on the values reported calculate the cost shares. Since the cost share of a user depends on her utility, the user may lie about her utility and lower her cost share, thereby increasing her welfare. Such users or agents are rational (*i.e.*, their goal is to maximize their welfare) and their behavior may not lead to a system wide optimal outcome.

Algorithmic Mechanism Design [8, 13] studies mechanisms where the outcome and the payment depend on the input provided by the participants. The task of the system designer is to design mechanisms that achieve system-wide goals in the presence of selfish agents. Mechanism design theory uses incentives to help designers achieve this objective. One important class of mechanisms are the so called *strategyproof mechanisms* [10]. In such mechanisms the participants obtain maximum profit when they declare truthfully their private values. In the standard AMD [13], a centralized trusted entity implements the mechanism, *i.e.*, it collects the input from agents, calculates the outcome and distributes the payment. When the mechanism is implemented in a distributed system [7], the agents themselves execute the algorithm and collectively calculate the outcome and the payments. Distributed implementations of mechanisms have been proposed for other problems such as divisible load scheduling [3, 4].

Feigenbaum *et al.* [7] proposed two distributed mechanisms to calculate the cost shares of the participants in a multicast transmission: the Marginal Cost (MC) mechanism and the Shapley Value (SH) mechanism. MC is a two phase mechanism whereas SH is an iterative mechanism. The convergence and scalability of these mechanisms have been studied theoretically, but no experimental evaluation was performed. These mechanisms assume that the agents may lie about their private values but they may not deviate from the specified distributed algorithm. This model of distributed implementation is known as the *Tamper-Proof Model (TPM)*. Thus we use MC-TPM and SH-TPM to denote the protocols proposed in [7]. However, this assumption is weak because agents can easily manipulate the distributed algorithm in their favor, since they control it. The model in which the agents may also deviate from the specified distributed algorithm, is called the *Autonomous Nodes Model (ANM)*. Mitchell and Teague [11] proposed an MC mechanism assuming the ANM. They augmented the original MC-TPM mechanism proposed in [7] with asymmetric key cryptographic primitives to prevent cheating. Specifically they used digital signatures to authenticate the sender of the messages and auditing to verify that the agents executed the protocol correctly. A Shapley Value mechanism for the ANM model was proposed and investigated in [9]. Similarly to [11], it uses digital signatures to authenticate the messages sent by the nodes. Auditing and verification is used to detect cheating by the nodes. The limitation in [11] is that it assumes that there is only one user per node, however, in [9] more than one user per node is allowed.

The MC mechanism is strategyproof and efficient (maximizes the overall welfare), but not budget balanced. In fact it is known that it generally runs budget deficit and in many cases does not generate any revenue at all [12], which is not favorable from the content provider's point of view. The MC mechanism is recommended when the multicast delivery may be subsidized [8]. The MC mechanism is also susceptible to collusion [5]. The SH mechanism is a better alternative from these considerations because it is budget balanced and group strategyproof. Although the SH mechanism is not efficient, for large user populations it approaches perfect efficiency. Also from the class of group-strategyproof mechanisms which are budget-balanced, the SH mechanism minimizes the worst-case welfare loss [12]. The only drawback of the SH mechanism is that it has a higher network complexity [6] and has been considered to be impractical. However there is no available experimental evaluation of the performance of this mechanism in the literature.

In this paper we investigate experimentally the performance of MC and SH mechanisms. We also study the behavior of the mechanisms from an economic perspective. We implement and deploy the mechanisms in a distributed real-world setting (PlanetLab). We run experiments to analyze the execution time of the mechanisms and the convergence of SH mechanism. We also investigate the number of users that receive the transmission and how much payment the content provider receives in different mechanisms.

The organization of the rest of the paper is as follows. In Section 2 we present the Marginal Cost and the Shapley Value mechanisms for both Tamper-Proof Model and Autonomous Nodes Model. Section 3 describes our experimental setup. Section 4 discusses the results of our experiments. We conclude the paper with a summary and directions for future work.

2. Cost Sharing Mechanisms

In this section we present the Marginal Cost and the Shapley Value mechanisms for the Tamper-Proof and the Autonomous Nodes models. The following network model and notation is used in the rest of the paper. There is a set of users P residing at N nodes. A node $i \in N$ is occupied by one or more users. The nodes are connected by bidirectional links creating a multicast tree. Creation of the multicast trees is discussed in [7, 14]. The transmission starts from a node $root \in N$ and flows through the static multicast tree. If a user e receives the transmission then $\sigma_e = 1$, otherwise $\sigma_e = 0$. The $root$ receives payment $payment_i$ from node i and also verifies if the payment received is correct (in ANM protocols). Each link connecting node i to its parent node p has a cost c_i associated with it. $\mathcal{C}_i$ denotes the set of all t children $k_1, \ldots, k_t$ of node i and r_i denotes the set of all the users at node i. In presenting the mechanisms, we use the primitive $\mathbf{send}(M, R)$ to denote the sending of message M to a node R and $\mathbf{recv}(M, R)$ to denote the receiving of message M from a node R.

2.1. Marginal Cost Mechanism for the Tamper-Proof Model (MC-TPM)

We briefly describe the additional notation used in this subsection and then present the mechanism. We denote the sum of utilities of all the users at node i

by $U_i = \sum_{e \in r_i} u_e$. The vector of utilities of all users $e \in P$ is denoted by u. We denote the subtree rooted at node i as T_i. T_i^+ denotes the union of T_i and the link from i to its parent p. $W(u)$ represents the net-worth of the system at utility vector u and $W_i(u)$ denotes the welfare (*i.e.*, utilities minus cost) of subtree T_i^+.

The mechanism (presented in Figure 1) is executed in two phases, a bottom-up phase and a top-down phase. In the bottom-up phase, the welfare values W_i are calculated by $W_i = U_i + \sum_{k \in \mathcal{C}_i | W_k \geq 0} W_k - c_i$.

Each node calculates the welfare value and sends it to its parent. Finally the root node receives the welfare values from all its children. Let A_i denote the smallest welfare value $W_{i'}(u)$ of any node i' in the path from i to *root*. A_i is used to decide which users receive the transmission and what will be their cost share. In the top-down phase, the minimum welfare value A_i of a node i is calculated and propagated down the tree. The details of how to calculate A_i and x_e are described in Figure 1. The properties of the mechanism are presented in [7].

```
Node i executes
Phase 1 (Bottom-up)
    for each child k ∈ C_i
        recv(W_k, k);
    Calculate W_i = U_i + Σ_{k∈C_i} W_k − c_i;
    if(W_i ≥ 0)
        set σ_e = 1 for all e ∈ r_i
        send(W_i, p);
    else
        set σ_e = 0 for all e ∈ r_i
        send(0, p);

Phase 2 (Top-down)
    if (node is root)
        for each child k ∈ C_root
            send(W_root, k);
    else
        recv(A_p, p);
        if(σ_e = 0 for all e ∈ r_i OR A_p < 0)
                x_e = 0, σ_e = 0 for all e ∈ r_i
            for each child k ∈ C_i
                send(−1, k);
        else
            Calculate A_i = min(A_p, W_i);
            for each child e ∈ r_i
                if(u_e ≤ W_p)
                    x_e = 0;
                else
                    x_e = u_e − A_p;
            for each child k ∈ C_i
                send(A_i, k);

Calculate payment_i = Σ_{e∈r_i} x_e;
send(payment_i, root);
```

Figure 1. MC-TPM mechanism

2.2. Shapley Value Mechanism for the Tamper-Proof Model (SH-TPM)

The Shapley Value mechanism for TPM is presented in [7]. This mechanism has two phases, a bottom-up phase and a top-down phase which are executed iteratively until the mechanism converges. Let n_i represent the number of users at node i who choose to receive the transmission. The bottom-up traversal is as follows. The bottom-up traversal starts from the leaf nodes, for which $\alpha_i = n_i$. Leaf nodes send α_i to their respective parents. Each node i receives α_k from all of its children. It then determines the number of users α_i in T_i who choose to receive the transmission using the following equation.

$$\alpha_i = \sum_{k \in \mathcal{C}_i} \alpha_k + n_i \tag{1}$$

After *root* receives $\alpha_i, i \in \mathcal{C}_{root}$, it initiates the top-down traversal, where it sends $\beta_{root} = 0$ to each of its children. β_i denotes the cost share of each of the resident users at node i, *i.e.*, $x_e = \beta_i, \forall e \in r_i$. Each node receives β_p from its parent and computes β_i as follows:

$$\beta_i = \left(\frac{c_i}{\alpha_i}\right) + \beta_p \tag{2}$$

β_i is then sent to all the children of node i (*i.e.*, all nodes $k \in \mathcal{C}_i$). All users $e \in r_i$ are assigned the cost share $x_e = \beta_i$. If the cost share x_e of any user e is greater than its utility u_e then user e declines to receive the transmission. In that case α_i decreases and it needs to be updated in the next bottom-up traversal. This decrease in α_i increases the cost shares of the other users sharing the links with e. Thus in each iteration of the bottom-up and the top-down traversal, users may be removed from the receiver set R and the cost shares updated. These iterations are repeated until no more users are dropped (which means that the cost share of any user does not change in two subsequent iterations). Initially $R = P$ and in the worst case one user is dropped in each iteration. If we assume that the algorithm converges in m iterations, the number of messages required in this case is $\Omega(n \times m)$. The mechanism is given in Figure 2. The following additional notation is used to describe the SH-TPM mechanism: j is the iteration number; α_i^j is the number of users in T_i who choose to receive the transmission in iteration j; β_i^j is the cost share of each user in the subtree rooted at node i; and n_i^j is the number of users at node i who

```
Node i executes
j = 0;
do
{
    Phase 1 (Bottom-up)
    j = j + 1;
    for each child k ∈ C_i
        recv(α_k^j, k);
    Calculate α_i^j = Σ_{k∈C_i} α_k^j + n_i^j;
    send(α_i^j, p);

    Phase 2 (Top-down)
    if (node is root)
        for each child k ∈ C_root
            send(0, k);
    else
        recv(β_p^j, p);
        Calculate β_i^j = (c_i/α_i^j) + β_p^j;
        for each child k ∈ C_i
            send(β_i^j, k);
} while (β_p^j ≠ β_p^{j−1});
Calculate payment_i = β_i^j * n_i^j;
send(payment_i, root);
```

Figure 2. SH-TPM mechanism

receive the transmission in iteration j.

2.3. Marginal Cost Mechanism for the Autonomous Nodes Model (MC-ANM)

Both mechanisms described above assume that the users may lie about their utility but the nodes do not deviate from the specified distributed protocol. Mitchell and Teague [11] questioned this assumption and considered that the nodes are autonomous and they can deviate from the algorithm to increase their welfare. To detect such cheating they proposed the use of digital signatures to authenticate the messages sent by a node and auditing by the content provider. The content provider ($root$) is also assumed to be the administrator of the mechanism. The administrator audits nodes with a given probability ($Prob_p$) and enforces a penalty in case of discrepancies.

The MC mechanism assuming ANM (MC-ANM) [11] is as follows. Assume that a message M signed using the private key K_i of node i is denoted by $E_{K_i}(M)$. The bottom-up traversal is the same as in the algorithm for TPM, except that the values sent by the children are signed using their private keys and each node verifies the signature after receiving the message. In the top-down phase each node i sends the message $E_{K_i}(A_i, W_k)$, where A_i is the value in the original protocol and W_k is the message which i received from child k during the bottom-up phase. At the end of the protocol the content provider audits each node i with probability $Prob_p$. It asks node i to send a proof of paying, $proof_i$ which consists of all the messages received by node i from its children and parent during the protocol. To check the proof, the content provider decrypts all the messages contained in $proof_i$ (which are signed by the parent and the children of node i) and calculates the utility and the payment expected from node i. If the actual payment received from node i is different from the expected payment, node i has to pay a high penalty, which prevents users from deviating from the protocol.

2.4. Shapley Value Mechanism for the Autonomous Nodes Model (SH-ANM)

The Shapley Value mechanism for the Autonomous Nodes Model (SH-ANM) is similar to SH-TPM but has to perform an extra auditing phase. The values sent by the nodes in the bottom-up and the top-down phases are calculated as in SH-TPM, however they are signed by the node before they are sent. Phase 1 (bottomup) and Phase 2 (top-down) are executed iteratively until the mechanism stabilizes. In each iteration j, in Phase 1, node i receives $E_{K_k}[\alpha_k^j]$ from each children $k \in \mathcal{C}_i$. Node i calculates α_i^j using (1) and sends $E_{K_i}[\alpha_i^j]$ to its parent p. Phase 2 (top-down) is initiated by $root$ sending $\beta_{root}^j = 0$ to its children. Node i receives $E_{K_p}[\beta_p^j \,||\, \alpha_i^j]$ from its parent p. It calculates β_i^j using (2) and sends $E_{K_i}[\beta_i^j \,||\, \alpha_k^j]$ to all its children. The cost share of the users at node i who receive the transmission is β_i^j.

These two phases are executed until the mechanism stabilizes (*i.e.*, no user drops from the receiver set), after which Phase 3 starts. In this phase the nodes send payments to $root$ and auditing is done by $root$. Assume that the mechanism stabilizes in m iterations. The payment sent by node i to $root$ is calculated using $payment_i = \beta_i^m * n_i^m$. The root node audits node i with probability $Prob_p$ by asking for a proof of payment: $proof_i = ||_{s=1}^m (n_i^s \,||\, E_{K_p}[\beta_p^s \,||\, \alpha_i^s] \,||\, E_{K_{k_1}}[\alpha_{k_1}^s] \,||\, \ldots \,||\, E_{K_{k_t}}[\alpha_{k_t}^s])$. It is assumed that the root knows the public key of all the nodes who want to participate in the multicast transmission. The root calculates α_i^j using (1) and β_i^j using (2) for each iteration j. It then verifies that the payment received from node i is $\beta_i^m * n_i^m$. If the payment received is not equal to $\beta_i^m * n_i^m$, node i has to pay a penalty $\mathcal{P}$ which is higher than any possible gain by cheating. More details on the mechanism and its properties are given in [9].

3. Experimental Setup

We implemented the mechanisms described in Section 2 in a distributed environment. We deployed the implementation on PlanetLab [2]. PlanetLab is a platform for developing, deploying and testing distributed services in a large scale distributed environment. We selected nodes randomly from all over the globe to obtain realistic data. The program implementing the mechanisms was run on all the nodes participating in the multicast transmission (including the root node). Messages are sent and received by the nodes to calculate the cost shares, demand/send proofs and to exchange control information (such as synchronization). The popular Openssl library [1] was used for encryption and decryption of messages (using public key cryptography, specifically RSA). We did not employ any key exchange mechanism, since we assume that the public keys of the child and the parent nodes are already available on each node.

For conducting our experiments we used different number of nodes (8 to 32) and correspondingly generated the multicast trees. The generated multicast trees are complete binary trees. The number of users per node was fixed in certain cases (mentioned below). In other cases the number of users at a node were randomly generated using the discrete uniform distribution over the interval [1,5]. The utilities of the users were also generated randomly using the uniform distribution over the interval [1,100]. We call a complete execution of a mechanism an experiment. Unless stated otherwise the probability of cheating by nodes is $Prob_c = 0.5$ and the probability of checking the proof is $Prob_p = 0.5$ (for the SH-ANM mechanism).

In order to study the convergence and the scalability of the mechanisms we varied the number of users per node and also the number of nodes (to vary the depth of the multicast tree). Since PlanetLab is a very dynamic environment and one experiment would be insufficient to draw conclusions, we ran 50 experiments for each set of values and reported the average over those experiments. The main data collected for analysis are the time required for each node to execute the mechanism, the number of rounds required to stabilize (for SH mechanisms), the number of users receiving the transmission and the payment received by the root.

4. Experimental Results

Figure 3 shows the execution time of the four mechanisms at each node for a multicast tree of sixteen nodes. The two SH mechanisms require more time than the two MC mechanisms. This is because they

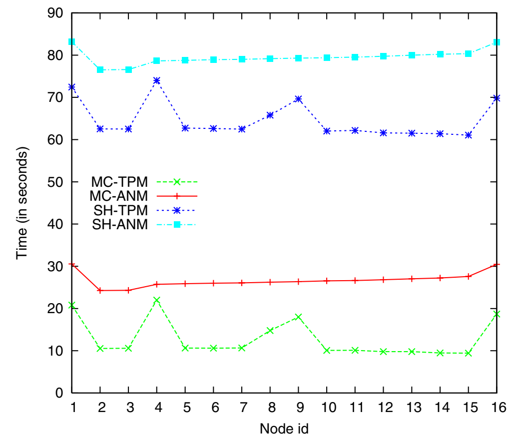

Figure 3. Execution time of the mechanisms

execute the bottom-up and top-down phases iteratively, whereas the MC mechanisms execute them only once. ANM mechanisms require more time compared to TPM mechanisms because of the additional messages for sending the proof and for checking. The spikes in the execution time of TPM mechanisms at node 4, 8 and 9 are due to the fact that the multicast tree used for the experiments is not balanced. The multicast tree is a complete binary tree with just one node (node number 16) at depth 4. Since messages are sent from children to the parent in the bottom-up phase (starting from leaf nodes until the message reaches the root) and similarly for the top-down phase, the depth of the tree plays an important role. Since node 16 is a child of node 8 which is in turn a child of node 4, the extra depth increases the time required by nodes 4 and 8 (which also affected node 9) to complete the execution. We believe that node 2 does not exhibit this increase in time because of its close proximity to the root. The ANM protocols do not exhibit the surges (at node 4, 8 and 9) because the proof-checking phase synchronize their finish time. We also observe that the time required by SH mechanisms is about three times greater than that of MC mechanisms. However this time is acceptable if the overall multicast transmission (generally a video or audio) is of several minutes. For a small transmission (like an advertisement), this time is not acceptable.

Another significant observation is with regard to the payment received and the number of users receiving the multicast transmission in the MC and SH mechanisms. As pointed out in [9] the MC mechanism is not budget-balanced and runs budget deficit in most of the cases, which is not a favorable property from the content providers perspective. In contrast, the SH mechanism is budget balanced. In Figure 4 we show

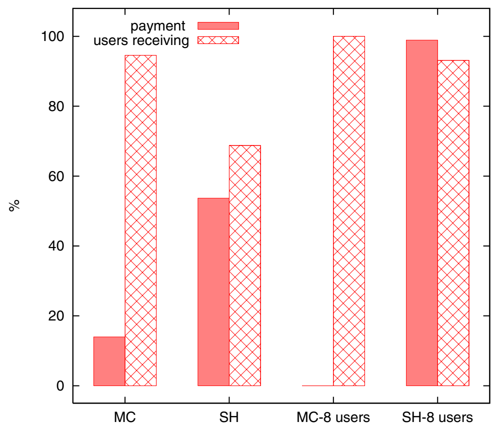

Figure 4. Payment received and number of users receiving transmission in MC and SH mechanisms

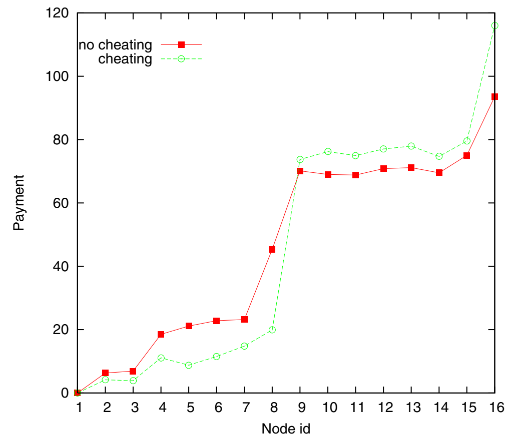

Figure 5. The effect of cheating in SH-TPM

the payment received as percentage of the total cost of all links. We denote by 'MC' the MC mechanism when the number of users per node is generated randomly within the interval [1,5] and by 'MC-8' the MC mechanism with 8 users per node. We use similar notation for the SH mechanisms. We observe from Figure 4 that the payment received in the case of MC mechanism is smaller than in the case of SH mechanism (for both [1..5] users and 8 users per node). MC-8 generated no revenue (all users pay 0), although all of them (100%) receive the transmission. In contrast SH-8 recovered more than 95% of the cost and about 98% users received the transmission. These results show that from the content provider's point of view, the SH mechanisms are more beneficial than the MC mechanisms.

As shown in [9], nodes can cheat in SH-TPM by sending manipulated values to their children, which causes the cheating nodes to pay less and the children nodes (of the cheating nodes) to pay more. We did controlled experiments to compare the effect of cheating. We used SH-TPM with 16 nodes and 8 users per node. Two cases were considered, one with no node cheating and another with nodes cheating with probability 0.5. In both cases the seed used to generate the random values was kept the same for a given node, so that the same utilities and link costs were generated. It is important to note here that only the nodes that have children have incentives to cheat [9]. Thus only nodes from 2 to 8 cheated. We observe in Figure 5 that the nodes that cheated paid less and the nodes who are leaf nodes paid more (to maintain the budget balance). We also observe (figure not shown here due to lack of space) that the percentage of users who receive the transmission increases for the cheating nodes (because when one node cheats all the users on that node receive the transmission irrespective of their utility). On the other hand, the leaf nodes suffer a decrease in the percentage of users receiving the transmission because their cost-shares increase and thus more users are dropped out at those nodes.

To study the effect of the number of users on the convergence of the SH mechanisms as well as other economic properties, we executed the SH-TPM protocol on 16 nodes fixing the number of users per node to 2, 4, 6 and 8 in different experiments. The utilities and link costs were generated randomly as explained above.

Counter-intuitively, the time required for the mechanism to stabilize decreases with the increase in the number of users, as shown in Figure 6. This is because of the interesting economics behind the mechanism. The users share the cost of each link and as the number of users increases the share per user decreases. Because of this, the mechanism stabilizes quickly and thus takes a smaller number of rounds to converge (as shown in Figure 7 - using the right vertical axis).

Another important consequence of increasing the number of users is that the total payment received by the mechanism increases as shown in Figure 7. This is because as the cost share per user becomes smaller, a smaller number of users drop out from receiving the transmission. Although the SH mechanism is budget balanced, it will not receive the maximum payment if a subtree of the multicast tree does not receive the transmission. In other words, the root will receive the maximum payment if every subtree of the multicast tree has at least one user receiving the transmission. This happens when there is a small cost share, which is obtained when the number of users is increased, as evident from the Figure 7. In the figure we plot the

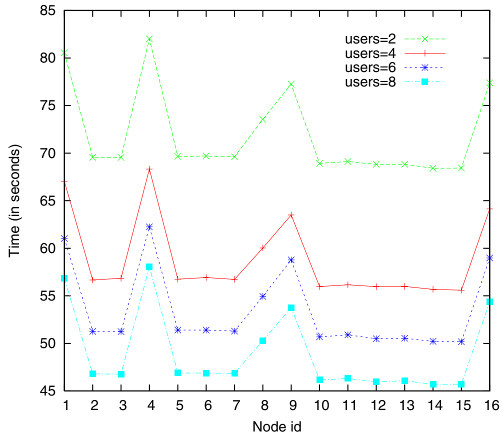

Figure 6. SH-TPM execution time at each node vs. number of users per node

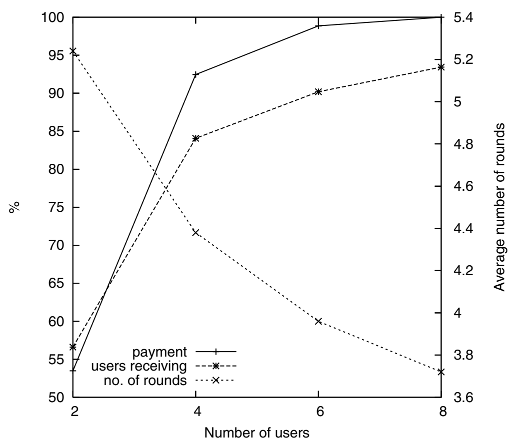

Figure 7. The effect of increasing the number of users per node in SH-TPM

ratio of the total payment received by the root to the sum of the costs of all links in the tree (not just those links which are involved in the transmission).

Another interesting property of SH-ANM mechanism is observed in Figure 8. We see that there is a decrease in the percentage of users receiving the transmission for nodes closer to the leaf nodes. This is because at the top of the tree, the cost share per user is small (because of a small number of links to connect to the root). Since the cost share is small, more users will be able to receive the transmission, even if their utilities are low. As we go toward the leaf nodes (as the number of links to reach the root increases) the cost share increases, thus reducing the number of users

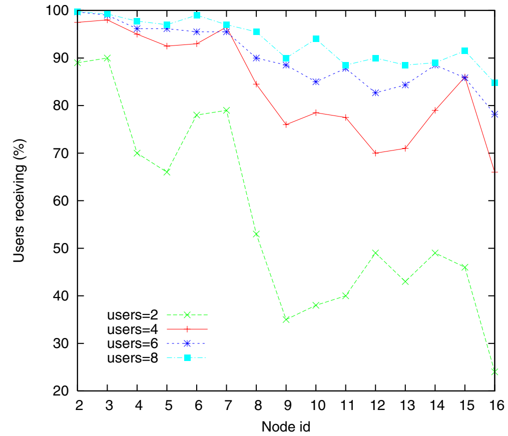

Figure 8. Percentage of users per node receiving transmission in SH-ANM

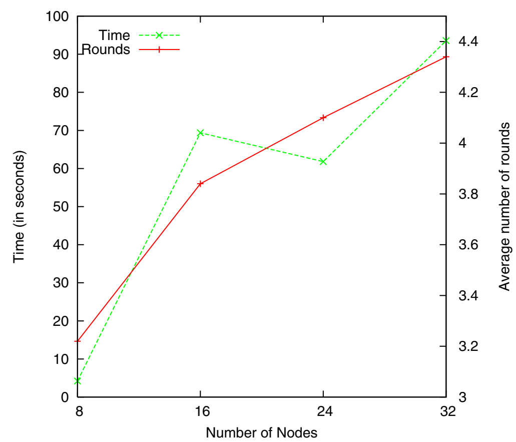

Figure 9. Execution time and number of rounds vs. number of nodes for SH-ANM

on those nodes who receive the transmission. It can be noted from Figure 8, that the nodes at the same depth have almost similar percentage of users receiving transmission. For example, for two users per node, in average about 90% of users at nodes 2 and 3 receive the transmission, while only 23% of users at node 16 receive the transmission.

In order to study the scalability of the mechanisms we conducted another set of experiments by varying the number of nodes participating in the transmission. For these experiments we used the SH-ANM mechanism and we kept fixed the number of users per node (8 users per node). $Prob_p$ and $Prob_c$ were both set to 0.5. As we see from Figure 9, the time required by the mechanism is proportional to the depth of the multicast tree, because of the nature of the mechanism (as explained above). We observe that when the number of nodes are increased from 16 to 24, the depth of the tree is the same and thus there is no increase in the execution time. The small decrease in time observed for 24 nodes is due to the nature of PlanetLab on which the application was deployed (experiments run at different time and different nodes). We also see from Figure 9 (the right side of the vertical axis) that the average number of rounds required to stabilize the mechanism increases with the increase in the number of nodes. This is quite opposite to the observation when the number of users was increased (there the number of rounds to converge decreased). The reason for this is that as the depth increases the cost share per user increases (especially for the leaf nodes), thus more users drop off, increasing the number of rounds necessary to converge. We also observe that the total number of users receiving the transmission decreases a little as the number of nodes is increased.

5. Conclusion

We implemented the Marginal Cost and Shapley Value mechanisms for sharing the cost of multicast transmissions on PlanetLab and studied their properties. We compared the MC and SH mechanisms for both TPM and ANM models in terms of their execution time. We also studied the convergence and the scalability of the mechanisms by varying the number of nodes and the number of users per node. We found that the SH mechanisms are slower than the MC mechanisms and that the TPM mechanisms are faster than ANM mechanisms. We showed that the MC mechanism generates a smaller revenue compared to the SH mechanism and thus it is not favorable for the content provider. From the computational point of view as well as economic considerations, increasing the number of users per node is beneficial for the system. As the number of users increases, the cost share per user decreases and thus more users receive the transmission and the mechanism converges quickly. This also works in favor of the provider because the revenue is close to the maximum when the number of users increases. However on the contrary increasing the depth of the multicast tree (by increasing the number of participating nodes) increases the time required to converge and also decreases the total number of users who receive the transmission.

In the future we would like to investigate the design of mechanisms that employ only incentive and do not use cryptography to prevent the cheating in the autonomous nodes model.

References

[1] Openssl. www.openssl.org.

[2] Planet-lab. www.planet-lab.org.

[3] T. E. Carroll and D. Grosu. Distributed algorithmic mechanism design for scheduling on unrelated machines. In *Proc. of the 8th Intl. Symp. on Parallel Architectures, Algorithms, and Networks (I-SPAN 2005)*, pages 194–199. IEEE CS Press, December 2005.

[4] T. E. Carroll and D. Grosu. A strategyproof mechanism for scheduling divisible loads in tree networks. In *Proc. of the 20th IEEE Intl. Parallel and Distributed Processing Symp. (IPDPS 2006)*, April 2006.

[5] J. Feigenbaum, A. Krishnamurthy, R. Sami, and S. Shenker. Approximation and collusion in multicast cost sharing. In *EC '03: Proc. of the 4th ACM Conf. on Electronic Commerce*, pages 280–280, June 2003.

[6] J. Feigenbaum, A. Krishnamurthy, R. Sami, and S. Shenker. Hardness results for multicast cost sharing. *Theoretical Computer Science*, 304(1-3):215–236, 2003.

[7] J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63(1):21–41, August 2001.

[8] J. Feigenbaum and S. Shenker. Distributed algorithmic mechanism design: Recent results and future directions. In *Proc. of the 6th ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 1–13, September 2002.

[9] N. Garg and D. Grosu. Faithful distributed shapley mechanisms for sharing the cost of multicast transmissions. *Technical Report, Dept. of Computer Science, Wayne State University*, September 2006.

[10] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory.* Oxford University Press, New York, 1995.

[11] J. Mitchell and V. Teague. Autonomous nodes and distributed mechanisms. In *Software Security - Theories and Systems. Mext-NSF-JSPS International Symposium (ISSS 2002)*, pages 58–83. Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2003.

[12] H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: budget balance versus efficiency. *Journal of Economic Theory*, 18(3):511–533, 2001.

[13] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behaviour*, 35(1/2):166–196, April 2001.

[14] S. Siachalou and L. Georgiadis. Algorithms for precomputing constrained widest paths and multicast trees. *IEEE/ACM Transactions on Networking*, 13(5):1174–1187, 2005.