Brief Announcement: Sustaining Collaboration in Multicast despite Rational Collusion

Haifeng Yu
National Univ. of Singapore
Republic of Singapore
haifeng@comp.nus.edu.sg

Phillip B. Gibbons
Intel Labs Pittsburgh
USA
phillip.b.gibbons@intel.com

Chenwei Shi
National Univ. of Singapore
Republic of Singapore
shichen@comp.nus.edu.sg

ABSTRACT

This paper focuses on designing incentive mechanisms for overlay multicast systems. Existing proposals on the problem are no longer able to provide proper incentives when rational users collude or launch sybil attacks. To overcome this key limitation, we propose a novel decentralized DCast multicast protocol and prove that it offers a novel concept of safety-net guarantee: A user running the protocol will always obtain at least a reasonably good utility despite the deviation of any number of rational users that potentially collude or launch sybil attacks.

Categories and Subject Descriptors: C.2.4 [**Computer-Communication Networks**]: Distributed Systems – *distributed applications*

General Terms: Algorithms, Design, Security

1. INTRODUCTION

In *p2p multicast systems* (e.g., Adobe Flash Player 10.1 and PPLive online TV platform), rational/selfish peers are supposed to help forward/relay the multicast data to other peers. This paper focuses on a key challenge in these systems, namely, how to incentivize these peers and sustain the collaboration. Similar to Equicast [2] and BAR gossip [4], we will consider a simple gossiping paradigm for p2p multicast. Here the multicast *root* is the source of the multicast data. A *user* has one or more identities (i.e., we allow sybil attacks), and each identity is called a *peer*. The gossiping process proceeds in synchronous *rounds*. In each round, the root sends (new) *multicast blocks* to some small number of randomly selected peers, while each peer selects some other peer from whom to pull (existing) multicast blocks. Each multicast block contains some fixed number of *multicast bits*.

Peers are rational/selfish and aim to maximize their *utilities*. Receiving more multicast bits increases the utility, while sending more bits or receiving more non-multicast bits decreases the utility. We assume that there exists some constant $\sigma > 1$ such that for any peer, the benefit of receiving σ multicast bits exceeds the cost of sending one bit or receiving one non-multicast bit. The multicast system provides a *protocol* (i.e., a strategy) to the peers. A peer may choose to follow the protocol or choose to deviate from the protocol in arbitrary ways, based on the utility achievable. A rational peer is called a *non-deviator* if it chooses to follow the specified protocol, otherwise it is called a *deviator*.

Previous results. Researchers have proposed several interesting and practical p2p multicast protocols [2, 3, 4] that eliminate profitable *individual* deviation and thus form Nash equilibria. On the other hand, their guarantee no longer holds when rational users collude, launch sybil attacks, or launch whitewashing attacks where a user abandons her/his identity to evade punishment and then rejoins with a new identity. Notice that sybil/whitewashing attacks can be viewed as a special case of collusion. More recently, Tran et al. [5] aims to maintain collusion-resilient reputation scores for peers, but their final guarantee is rather weak and colluding peers can increase their reputation scores unboundedly as the number of colluding peers increases.

Challenges. The inability of these previous approaches to deal with collusion is related to the following two challenges. First, the key to incentivizing collaboration is always a punishment mechanism to punish or reward less those peers who fail to collaborate. The presence of collusion makes it challenging to punish. Evicting a peer (or refusing to send data to that peer) as in [2, 3, 4, 5] is no longer effective — the evicted peer may obtain multicast data from its colluding peers. Moreover, with sybil attacks and whitewashing attacks, eviction simply has no effect on the user.

Second, in some cases the colluding peers may be able to obtain the multicast data from each other more efficiently. For example, suppose the protocol provided by the multicast system is based on gossiping for better robustness against churn. If the colluding peers have low churn, then they can switch to using more efficient tree-based multicast among themselves. Such deviation is already profitable. Furthermore, the colluding peers can either continue to gossip with the non-deviators as usual, or they can participate in gossiping with the non-deviators less frequently. Detecting such deviation, from the non-deviators' perspective, is rather difficult if not impossible.

Our goal. Given such context, the goal of this work is to design a p2p multicast protocol that can properly sustain collaboration despite collusion and sybil/whitewashing attacks by rational users.

2. SAFETY-NET GUARANTEE

The natural way to capture rational collusion is to use the concept of various collusion-resistant Nash equilibria. However, our example earlier already hints that unless a protocol offers optimal performance (i.e., minimizing the overheads incurred by sending/receiving bits) for each possible subset of the peers (without knowing their specific properties such as low churn rate), some subset can *always* profit by switching to a more optimized protocol. Given such impossibility of preventing deviation, aiming to achieve collusion-resistant Nash equilibria would simply be futile. Fortunately, preventing deviation is not actually necessary to sustain collaboration. After all, deviation by itself is not harmful—it is the deviation's negative impact on other (non-deviating) peers that is harmful. This

[1] This work is partly supported by Academic Research Fund grant R-252-000-406-112.

PODC'11, June 6–8, 2011, San Jose, California, USA.
ACM 978-1-4503-0719-2/11/06.

basic observation leads to to our novel concept of a *safety-net guarantee*, which formalizes the goal of this work.

We say that a protocol offers the *safety-net guarantee* if for any peer A that chooses to follow the protocol, A obtains at least a reasonably good utility (called the *safety-net utility*), despite *any* set of colluding peers deviating from the protocol using a *pareto-optimal strategy profile*. We require the collusion strategy to be pareto-optimal since the colluding peers are rational (see [6] for details). The safety-net guarantee is not concerned with protecting the utility of the deviators — if a deviator's utility is below the safety-net utility, it can always switch back to being a non-deviator. We emphasize that the safety-net guarantee does not prevent deviation. In the extreme, for a protocol offering the safety-net guarantee, it is possible for *all* peers to deviate from that protocol.

Our safety-net guarantee is related to the *price of collusion* [1], which quantifies the negative impact of collusion on the *overall* social utility in a congestion game. In comparison, the safety-net guarantee bounds the negative impact of collusion on the utility of *individual* non-deviators in a multicast game. Furthermore, we consider all pareto-optimal strategy profiles of the colluding peers, while the price of collusion focuses on one particular pareto-optimal strategy profile (i.e., the one maximizing the sum of the utilities).

3. DCAST PROTOCOL

Having introduced the safety-net guarantee, we now propose a novel and elegant *DCast* protocol, which is the first practical multicast protocol achieving such guarantee. This section focuses on the overview and intuition — see [6] for the detailed protocol, pseudocode, theorems, proofs, and implementation. We assume that the multicast session has an infinite number of rounds to avoid the well-known end-game effect in finite-horizon repeated games (see [6] for how this assumption can be weakened).

Overview of DCast. Section 1 described a simple pull-based gossiping paradigm for p2p multicast. In this paradigm, colluding peers can profitably deviate from the protocol in several ways. For example, a colluding peer A can pretend that it has no multicast blocks to offer when a non-deviator pulls from A. A can also pull from multiple non-deviators in each round. A user may further launch a sybil attack to attract more multicast blocks directly from the root. DCast builds proper incentives into such pull-based gossiping so that each such deviation *either* is non-profitable *or* will not bring down the utilities of the non-deviators below the safety-net utility. In designing the incentives, DCast addresses the two challenges discussed in Section 1 via the novel design of *debt-links* and *debt-coins* (or *doins* in short).

During the pull-based gossip in DCast, the propagation of a multicast block from one peer A to another peer B is always coupled with the propagation of a *doin* on an unoccupied *debt-link* from A to B. A *debt-link* from A to B is established by B sending $\sigma + 1$ *junk blocks* to A. A junk block contains only junk bits and is of the same size as a multicast block. Notice that establishing the debt-link hurts the utility of both A and B. A debt-link is *unoccupied* when first established. After propagating a doin via that debt-link, the debt-link becomes *occupied* until the corresponding doin is *paid*. A doin is a debt and can be *issued* by any peer. The current holder of a doin conceptually "owes" the issuer of the doin. Doins may circulate (i.e., be relayed) in the system and thus can be viewed as a special kind of bankless virtual currency. Doins will expire every fixed number of rounds, after which point new doins will be issued. A peer holding an expired doin will pay for that doin by sending the doin issuer σ multicast blocks.

Debt-links as *pairwise* entry fees. Fundamentally, the debt-links established by a peer in DCast are *pairwise entry fees* paid by that peer. In other words, the peer incurs some bandwidth consumption to be allowed to interact with some specific peers (i.e., those peers from which the debt-links are established) in a limited form (i.e., the peer cannot borrow more multicast blocks than the number of unoccupied debt-links). This entry fee is pairwise instead of system-wide in the sense that the peer is not allowed to interact with all peers in the system. The pairwise nature prevents a colluding peer from giving other colluding peers interaction access to non-deviators.

The above entry fee serves as an effective punishment to a peer that fails to pay for a doin, since the cost of paying the doin is smaller than the entry fee itself. This is true even in the presence of collusion and sybil/whitewashing attacks. A colluding peer A may be able to obtain multicast blocks from other colluding peers. But if A does establish an incoming debt-link from another peer B, it indicates that A is not able to rely on other colluding peers only, and has to seek B's help. Given that doins are eventually paid, if A pulls from more than one non-deviator in a round, those non-deviators will get payment later and their utilities will be properly protected.

Making doin issuance/relay profitable. The implicit entry fee associated with debt-link establishment from A to B is in the form of junk blocks, which is necessary since a new peer B has no useful data to offer as entry fee. On the other hand, with this design A actually has disincentive to accept debt-link establishments. DCast solves this problem by setting the doin payment amount to be σ and by properly re-using debt-links. Under such payment amount, A makes some (constant) profit each time a doin is issued/relayed on a link and then paid. Even for colluding peers who may enjoy a more optimized (e.g., tree-based) protocol among themselves, we expect a σ of 2 or 3 will be sufficient to make them a profit. Re-using the debt-link a sufficient number of times during the multicast session will then enable the accumulated profit to exceed the initial setup cost of the debt-link. This in turn, incentivizes colluding peers to send multicast blocks when non-deviators pull from them.

Root sending blocks to peers. Finally, peers do not establish any debt-links from the root. Before the root sends a multicast block to a peer, the peer is required to send $\sigma + 2$ junk blocks to the root. This provides disincentive for a rational user to launch a sybil attack in order to attract more multicast blocks from the root.

Formal guarantees. We are able to prove [6] that DCast offers a safety-net guarantee under some reasonable conditions. Roughly speaking, the safety-net utility offered by DCast is such that with high probability, a non-deviator obtains all multicast data needed while sending $(\sigma + 2)(1 + \rho)$ bits for each multicast bit received. Here ρ is a constant describing the relative number of control bits in the protocol as compared to the number of multicast/junk bits.

4. REFERENCES

[1] A. Hayrapetyan, E. Tardos, and T. Wexler. The effect of collusion in congestion games. In *STOC*, 2006.

[2] I. Keidar, R. Melamed, and A. Orda. EquiCast: Scalable multicast with selfish users. In *PODC*, 2006.

[3] H. C. Li, A. Clement, M. Marchetti, M. Kapritsos, L. Robison, L. Alvisi, and M. Dahlin. Flightpath: Obedience vs. choice in cooperative services. In *OSDI*, 2008.

[4] H. C. Li, A. Clement, E. L. Wong, J. Napper, I. Roy, L. Alvisi, and M. Dahlin. BAR Gossip. In *OSDI*, 2006.

[5] N. Tran, J. Li, and L. Subramanian. Collusion-resilient Credit-based Reputations for Peer-to-peer Content Distribution. In *NetEcon*, 2010.

[6] H. Yu, P. B. Gibbons, and C. Shi. DCast: Sustaining Collaboration despite Rational Collusion. Technical Report TRA2/11, School of Computing, National University of Singapore, Feb 2011. Available at `http://www.comp.nus.edu.sg/~yuhf/TRA2-11.pdf`.