1. MOTIVATION

Why is the problem addressed in the paper interesting and important for the larger community to be solved?

The problem is that in conventionally tree-based multicast system, the burden of forwarding as well as duplicating multicast messages are only taken care of by interior nodes, however those nodes are small subset, and if they are not highly available nor dedicate organized, the system doesn't work. On the other hand, the majority, those leaf nodes, contribute no resources to forwarding load, wasting their outband bandwidth. What's more, once the tree grows deep, the system is not practical, since it becomes fault prone and introduces large delay.

2. CONTRIBUTION

What are the main contributions of the paper?

The main contribution of the paper is to introduce SplitStream, presenting a forest of interior-node-disjoint multicast trees that enables distribution of high-bandwidth content with application-level multicast in a cooperate environment. The goal is to make all peers contribute resources in order to distribute the forwarding load. Another point of the paper is the idea of splitting the multicast stream into multiple stripes with separate multicast trees based on a non-dedicated infrastructure, so that it accommodates different limitations, since each node may have different bandwidth capacities, and also increases the resilience to node failure and sudden node departures.

3. SOLUTION

How did the authors solve the problem at hand?

As the name SplitStream indicates, the original stream is split into k stripes, and there is a separate multicast tree for each stripe, and in order to achieve fault tolerance, application uses erasure coding or multiple description coding to mitigate the effects of node failure, the content is encoded in the way such that each stripe acquires nearly the same bandwidth and any subset of the stripes with adequate size can rebuild the content. The trees are interior-node-disjoint, which require nodes to be interior in at most one tree, in result node failure affects only one stripe, and the trees are built on Pastry's routing properties with setting $2^b = k$. The stripe ids differ on the first digits, and nodes are claimed to join at least the stripe which owns the same first digit as their own ID. Regarding to the bandwidth problem, the push-down mechanism in Scribe is furthermore tailored, such that adoption always happens, but a random child share no prefix in the set, or with the shortest prefix in common with the stripe id is rejected. The orphaned child tries to attach a parent among former siblings with proper prefix by push-down; otherwise it anycasts the space capacity group and performs depth-first search for a parent. Nevertheless, two conditions are necessarily to be held in forest construction: sum of desired indegrees is less than or equal to sum of forwarding capacities and nodes whose forwarding capacity is greater than desired indegree should receive or originate all k stripes.

4. EVALUATION

How good is the solution?

The solution is creatively, practically, efficiently and has high commercial value, especially in High-bandwidth environment. It also has an acceptable failure recover mechanism.

How did the authors evaluate their solution?

There are two goals of the evaluation, one is the measure of the overhead of forest construction in SplitStream, and the other is test of the performance of multicasts using SplitStream. They test their solution in two environments, a simulated network and a real testbed in the Internet. In the network simulation, they use three different network topologies, GATech, Mercator and CorpNet to simulate the propagation of delay in real network. Six different configurations are used in this stage. And four of them aim to test the impact on overhead as well as spare capacity. The other two aim to test with different desired indegrees and forwarding. While evaluate the construction, node stress and link stress are considered. And in the performance test, they compare node stress, link stress, and delay with IP multicast and Scribe. Node failures are evaluated separately in three experiments: path diversity, catastrophic failures and high churn. In the real internet environment, the Planet Lab, they run the SplitStream in 36 hosts and each of them has two nodes. They illustrate three figures which show the receiving of package, delay of package and CDF of the delays.

How good was the evaluation of their work?

The solution of evaluation is convictive to prove their algorithm's advantage. It is quite comprehensive. Since they test the SplitStream both in simulate environment and real network. The experiment of nodes failure is also significative.

5. DISADVANTAGES OF THE SOLUTION

What are the disadvantages and shortcomings of the solution given by the authors?

SplitStream works well under the assumption that in the communication the bandwidth bottleneck is at the sender or receiver, if the bottleneck is somewhere else, nodes may not be able to receive all stripes desired. When locating parents, it is possible for the node to append to a parent which is already an interior node in another stripe tree. Therefore if this parent fails, some nodes may lose more than one stripe in a temporary time. Anycasting to the spare capacity group may fail to locate a parent, for example, if the group is empty then an orphan remains after all forwarding capacity has been consumed. Also anycasting can fail if no node of the spare capacity group supports any of the desired stripes.

6. DISADVANTAGES OF THE EVALUATION

During the evaluation of their solution, did the authors overlook something?

In the large-scale network simulations, they did not consider the queuing delay, packet losses or cross traffic. In the PlanetLab experiment, they may as well involve more nodes that can show the advantage of SplitStream more significantly. And such applications in the real world usually have much more than just 72 nodes.

7. FURTHER IMPROVEMENTS

Are there any further improvements that can be made to the solution?

Nodes may keep track of the packet arrival rate for each stripe, once they find the incoming channel for a stripe is delivering the bandwidth behind expectation, they can break away from the stripe tree and look for an alternative parent.

Are there any future directions you can think of?

A mechanism is needed to discourage freeloader, so that the forwarding capacity of most participants is greater than or equal to its desired indgree, such mechanism may base on incentives or use a trusted execution platform.