Motivation

In traditional tree-based multicast system there are small number of interior nodes which burden the load of forwarding multicast messages on most of peers are leaf nodes and are not well matched to a cooperative environment, on the other hand by SplitStreasming, the load is distributed among nodes.

SplitStream is highly robust to failure because a node failure causes the loss of a single stripe on average.

Contribution

By splitting the content into stripes, SplitStream application makes separate trees for each stripe in order to divide the load of forwarding multicast messages overall peers. Constructing such forest makes this opportunity that the forwarding load is almost balanced across all nodes while achieving low delay and link stress over the system.

Striping to multiple trees also increases the resilience to node failures. SplitStream tries to ensure that huge numbers of nodes are interior nodes in only one tree. If failure of a single node causes the temporary loss of at most one of the stripes by providing an appropriate data encodings, applications can mask or mitigate the effects of node failures even while the affected tree is being repaired.

Solution

In SplitStream, multicast message is split in to stripes and a multicast tree is constructed for each stripe. So Participating peers may receive a subset of the stripes (which depends on their inbound bandwidth) and may have several children (depends on outbound bandwidth) for forwarding stripes. For the work load to be fairly divided each node should be an interior node for one tree and a leaf node for other multicast trees (interior-node-disjoint trees). One solution for how application level multicast trees can be built and maintained is exploiting Scribe (an application-level group communication system built upon Pastry) and Pastry (for routing).
Pastry ensures that the multicast trees are efficient. Because the low delay of Pastry route the delay for forwarding a message from the root to each group member is low. Pastry's local route convergence ensures that the load imposed on the physical network is small because most message replications occur at intermediate nodes that are close in the network to the leaf nodes in the tree. For not violating the outbound capacity the algorithm "push-down" handles the join request of children to nodes which have not outbound left, by pushing down the orphan it tries to find a parent, if it couldn't find any suitable parent it sends an anycast message to the spare capacity group (a group which all nodes in that group have less children than their outbound capacity). The recipient of message is responsible for finding the parent.

Evaluation

Distributing the work load fairly among all peers in a decentralized and scalable environment which reduces the bandwidth constraints on the nodes is very useful. Also by this method, recovering from failure is easier. However, constructing and maintaining the multicast tree is a complex task and the amount of memory which each node needs to maintain information for all the trees it belongs to is considerable.
For experimental evaluation a packet-level, discrete-event network simulator has been used. In the simulations, GATech, Mercator and CorpNet topology model has been exploited. An enhanced SplitStream implementation which its optimizations relate to cycle detection, reduction of anycast message and improvement of DFS traversal of spare capacity group tree is also introduced. The evaluation tries to show the overhead of forest construction in Split-Stream

and the performance of multicasts using the forest with and without failures**.** SplitStream is evaluated with six different configurations of node degree constraints. Four of them are designed to evaluate the impact on overhead and performance of varying the spare capacity in the system, and other configurations evaluate the behavior of SplitStream when nodes have different desired indegrees and forwarding capacities. The metrics for evaluation without node failures to measure the overhead are node stress (the number of messages that a node receives) which quantifies the load on nodes and link stress (number of messages sent over the link) which quantifies the load on the network are introduced. For the forest multicast performance the delay penalty relative to IP is introduced as a measure in forest performance.
For the evaluation with failures the metric for performance, the fraction of the desired stripes received by each node, and for the overhead metric, the number of control messages sent per second by each node are introduced.
The results for the forest construction overhead show that when the spare capacity increases the link and node stress increase. During forest Construction it is shown that node and link stress are low and node stress is not affected by the number of peers. It has been also shown that, the link stress and the delay penalty decrease when the spare capacity increases. If we compare to IP multicast, SpliStream uses much more links in the network.
It also has been shown (in the experiments with node failures) that SplitStream is highly tolerant of failures. Because of the diverse paths it constructs, the system manages to lose few stripes in case of failures. And it can be soon recovered even there is a high churn.

Disadvantages of solution

When the bandwidth bottleneck in the communication between two nodes is neither at the sender nor at the receiver may nodes be unable to receive all desired stripes.

Disadvantages of evaluation

It was better if the SplitStream be compared to other application level multicast systems like OverCast and CoopNet.
It was needed to experiment the evaluation with UDP, too.

Further improvements

According to the problem which is described in the disadvantages of solution, a solution could be as follow, nodes can monitor the packet arrival rate for a stripe, if they are not satisfied they can detach from the stripe tree and search for an alternate parent.