feedtree

Distributing Web news with peer-to-peer multicast

Dan Sandler, Alan Mislove,
Ansley Post, Peter Druschel
Rice University

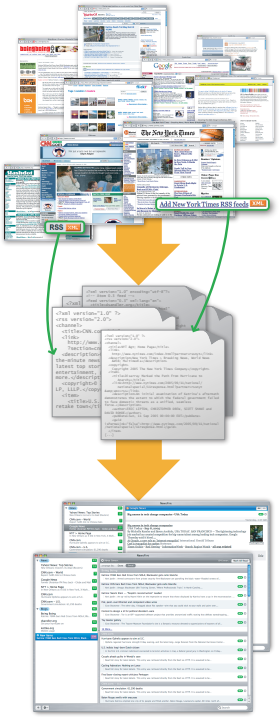
Background:

How RSS saved the Web.

The Web has experienced an explosion of MICRONEWS: highly focused chunks of content, published frequently and irregularly, scattered across scores of sites. Web surfers accustomed to clicking daily through one or two bookmarks are increasingly out of the loop.

RSS FEEDS have become a popular way to deal with this information flow. Alongside its usual HTML pages, a website may publish a summary of its most recent news stories in an XML-based format called RSS. (The availability of a site's feed is commonly advertised with an orange XML icon.)

USERS THEN "SUBSCRIBE" TO RSS FEEDS with special reader software, which periodically collects the latest items from the user's subscriptions and organizes them for convenient reading. At any time, a user can glance at her RSS reader to get a concise picture of the news she cares about. IT'S LIKE EMAIL FOR WEB NEWS, and it's proving very popular with users.



Problem:

RSS isn't scaling well.

Feed publishers have become concerned over the way in which RSS feed data is transferred over the network. RSS readers check for news by REPEATEDLY POLLING A NEWS FEED'S URL (typically once or twice per hour). Feeds can therefore consume more bandwidth than a typical Web resource. Some publishers have begun CURTAILING RSS SERVICE to cope.

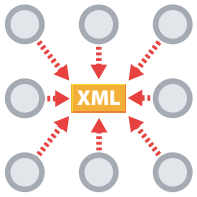
At the same time, end users want to see more timely news (that is, shorter delays between updates), so users have every incentive to *exacerbate* the stress on publishers by POLLING FEEDS EVEN MORE FREQUENTLY.

FeedTree:

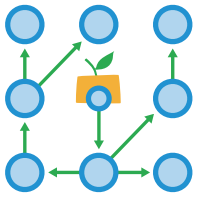
Cooperative micronews.

FeedTree addresses these problems by replacing the polling architecture of RSS with a PEER-TO-PEER (P2P) APPROACH. Users of FeedTree become "nodes" in PASTRY, A SELF-ORGANIZING P2P OVERLAY NETWORK developed at Rice. Rather than individually and redundantly polling a central server, nodes organize into MULTICAST TREES (one for each feed) to distribute new RSS data promptly and efficiently.

FeedTree-aware publishers INJECT NEW DATA IMMEDIATELY into the FeedTree network, eliminating the hour-long news delay of conventional RSS. Legacy feeds (those not multicast directly by the publisher) are polled by a subset of the nodes and shared with all subscribers.

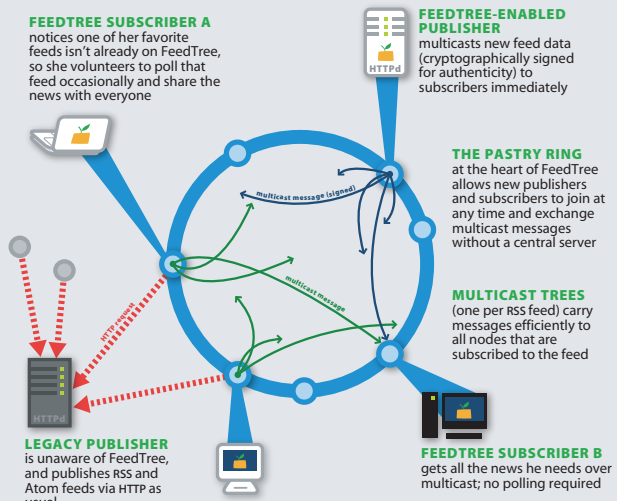


CONVENTIONAL DELIVERY
RSS readers poll the feed's Web server independently.



FEEDTREE DELIVERY
News "pushed" immediately by publisher. Nodes cooperate to forward news efficiently.

Participants in the FeedTree network.



FEEDTREE SUBSCRIBER A
notices one of her favorite feeds isn't already on FeedTree, so she volunteers to poll that feed occasionally and share the news with everyone

FEEDTREE-ENABLED PUBLISHER
multicasts new feed data (cryptographically signed for authenticity) to subscribers immediately

THE PASTRY RING
at the heart of FeedTree allows new publishers and subscribers to join at any time and exchange multicast messages without a central server

MULTICAST TREES
(one per RSS feed) carry messages efficiently to all nodes that are subscribed to the feed

LEGACY PUBLISHER
is unaware of FeedTree, and publishes RSS and Atom feeds via HTTP as usual

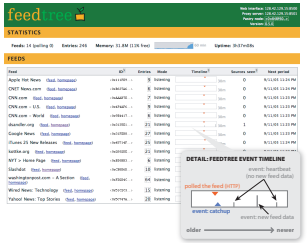
FEEDTREE SUBSCRIBER B
gets all the news he needs over multicast; no polling required

FEEDTREE SUBSCRIBER C
volunteers to poll A's feed too, on an offset schedule—effectively doubling that feed's update rate for FeedTree participants

Implementation:

Living in the real world.

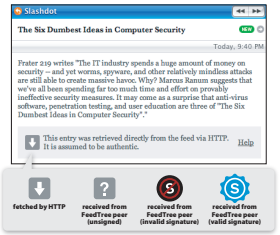
We have built an HTTP PROXY that brings the benefits of FeedTree to any existing desktop RSS reader. The *ftproxy* application becomes a node in the FeedTree network and waits for the user's RSS reader to request a feed (by making an HTTP request for the feed's URL). In response to this request, *ftproxy* joins the FeedTree multicast tree for that feed, and begins listening for pushed updates. *ftproxy* will respond to future requests for the same URL by substituting the most up-to-date FeedTree updates for that feed.



SCREENSHOT: THE FEEDTREE PROXY
Web-based monitoring interface gives an overview of recent FeedTree events.

Feed authenticity.

Peer-to-peer multicast means that FeedTree users receive events from untrusted peers. Therefore publishers are encouraged to push CRYPTOGRAPHICALLY SIGNED FEED DATA using the FeedTree publishing tool. The publisher's public signing key is included in the conventional RSS feed for FeedTree nodes to download and use when verifying received data.



SCREENSHOT: SECURITY FOOTER
FeedTree appends authenticity information to each RSS entry's contents; the user can read this footer in any HTML-enabled RSS software.

Conclusion:

Better RSS service for everyone.

The FeedTree software is available today from FEEDTREE.NET. Users running *ftproxy* will see BETTER SERVICE than conventional RSS polling can provide. Publishers who install the *ftpublisher* tool will ensure TIMELY, AUTHENTIC UPDATES to FeedTree users.

FeedTree also represents a REAL-WORLD APPLICATION OF PEER-TO-PEER RESEARCH, and presents an excellent opportunity to study and improve the performance of these algorithms on real users' desktops and under real workloads.

Available now at feedtree.net.